

Creating Interactive 3-D Graphics!

#267 JULY 1997

Dr. Dobb's

JOURNAL

SOFTWARE
TOOLS FOR THE
PROFESSIONAL
PROGRAMMER

<http://www.ddj.com>

GRAPHICS PROGRAMMING

- Real-Time 3-D Graphics for Modeling & Animation
- Your Own Cross-Platform Graphics Framework
- Special Effects for Realistic Images

IMPROVING WEB SERVER PERFORMANCE

DATA STRUCTURES & JAVA

SOUNDING OFF
WITH 3-D AUDIO

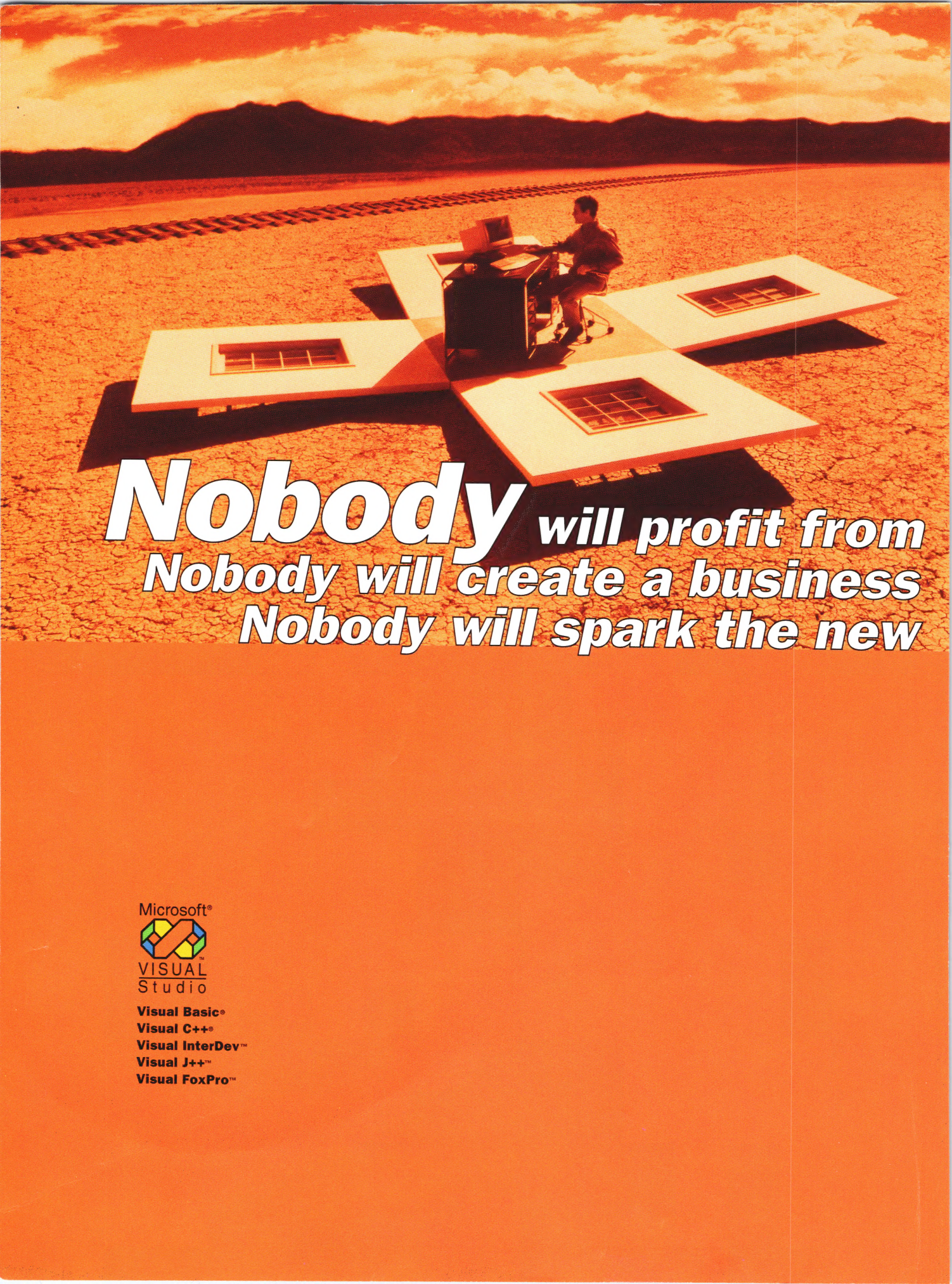
JAVA, VRML, C++,
BASIC, AND MORE!



\$4.95 (\$5.95 CANADA)



A Miller Freeman Publication



Nobody will profit from
Nobody will create a business
Nobody will spark the new

Microsoft®



VISUAL
Studio

Visual Basic®

Visual C++®

Visual InterDev™

Visual J++™

Visual FoxPro™



***a medium in its infancy.
without walls.
millennium.***

***(Until somebody has
the tools to
make it happen.)***

Microsoft® Visual Tools are uniquely designed to help you create the applications, the components, and the building blocks of the future. Now, Microsoft Visual Studio™ 97 provides a comprehensive set of tools to help you build solutions which integrate your client/server applications with Internet technologies. Visual Tools support the industry standards for components, data access, development languages, and the Web. For large-scale development, the Enterprise Edition of Visual Studio features Visual Database Tools, Visual Modeler,* Microsoft Repository, Visual SourceSafe™, and the Developer Editions of

Microsoft SQL Server™ and Microsoft Transaction Server. To ensure your success, Microsoft offers broad support from timely information with the Microsoft Developer Network to interactive training with the Microsoft Mastering Series. And with Office 97 Developer Edition, you'll get everything you need to integrate the Office client with your enterprise. To learn more about Microsoft Visual Tools, please visit us on the Web.

Microsoft®

Where do you want to go today?® www.microsoft.com/VisualTools/

© 1997 Microsoft Corporation. All rights reserved. Microsoft, Visual Basic, Visual C++, and Where do you want to go today? are registered trademarks and SQL Server, Visual FoxPro, Visual InterDev, Visual J++, Visual SourceSafe, Visual Studio, and the Visual Studio logo are trademarks of Microsoft Corporation. To order any of the Microsoft Visual Tools, or to receive a reseller referral, in the United States or Canada, please call (800) 621-7900, Monday through Friday (7:00 a.m. to midnight, Eastern time), Saturday (8:00 a.m. to 8:30 p.m.), and Sunday (8:00 a.m. to 7:00 p.m.). Outside the U.S. and Canada, please contact your local Microsoft subsidiary. TT/TDD available at (800) 892-5234.
*Available through Web download to registered users of Visual Studio Enterprise Edition.

FEATURES

IMPLICIT SURFACES AND REAL-TIME GRAPHICS

by Kyle Lussier

Implicit surfaces are used in molecular modeling, animation, geophysical systems, weather analysis, and the like. They're also at the heart of Silly Space, the modeling application Kyle describes here.

A MEMORY-CONSTRAINED IMAGE-PROCESSING ARCHITECTURE

by Mayur Patel

Image-processing applications are hungry for both processing time and memory. The architecture Mayur presents here puts you in the driver's seat when it comes to controlling memory consumption for high-performance image processing.

RAVEKIT: A PORTABLE GRAPHICS FRAMEWORK

by Mark Carolan

RaveKit, Mark's general-purpose, cross-platform framework, lets you create interactive 3-D environments that provide an alternative way of presenting graphical data and obtaining user input.

MOTION BLUR EFFECTS

by Tim Wittenburg

Altering the appearance of "moving" objects using blurring techniques can add realism and excitement to your graphic images. Tim shows you one way to accomplish this.

TREAPS IN JAVA

by Stefan Nilsson

Treaps, the randomized search trees devised by C.R. Aragon and R. Seidel, provide the functionality of a general-purpose sorting routine, priority queue, hash-table, stack, or standard queue. Stefan shows you how to implement treaps in Java.

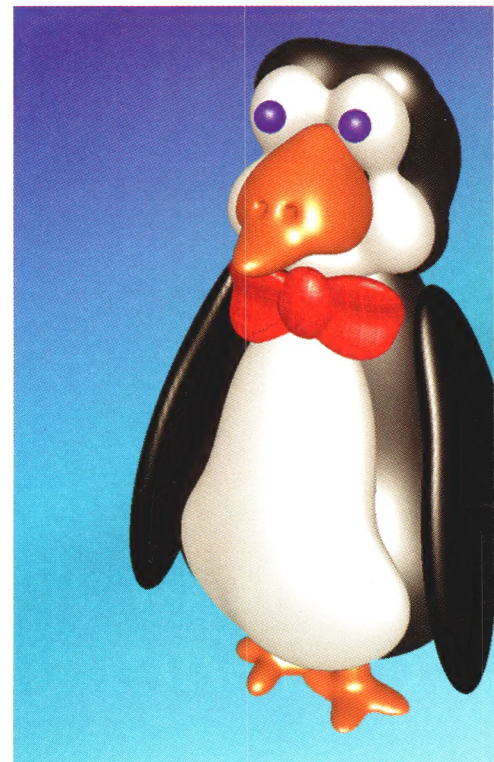
18

24

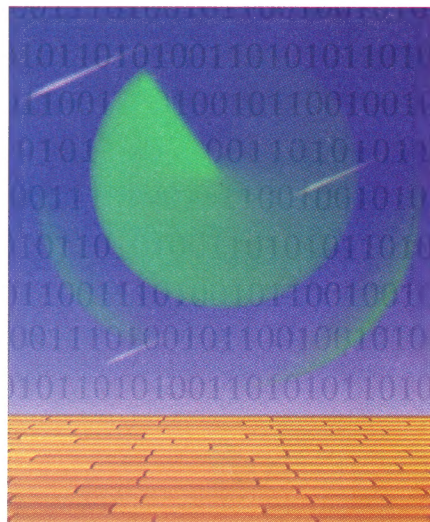
30

36

40



Thanks to Kyle Lussier for the penguin on our cover.



EMBEDDED SYSTEMS

ATMEL'S AT89C2051 MICROCONTROLLER

by Dhananjay V. Gadre

Dhananjay presents a simple programmer for AT89C2051 that is hosted on a 8052-based circuit running a Basic interpreter. He then uses the system to build a 12-bit multichannel ADC to connect to PCs.

46

INTERNET PROGRAMMING

THREAD POOLS AND SERVER PERFORMANCE

by John Calcote

As John points out here, thread pools provide one way of improving server performance.

60

PROGRAMMER'S TOOLCHEST

SOUNDING OFF WITH THE RSX LIBRARY

by Steve Durham

Intel's Realistic Sound Experience (RSX) library was initially developed to enhance the silent 3-D world of VRML. Steve presents WaveSpinner, an RSX-based tool that reads a script, animates and spatializes all the sounds using RSX, and writes the result to a .WAV file.

66

TECATE AND INTERACTIVE 3-D

by Peter D. Kochevar

By providing a set of tools for describing animated 3-D scenes and their interaction with end users, the Tecate system enables the description of interactive, animated 3-D scenes, and interfaces to existing applications and data sources such as the World Wide Web and remote collaboration.

72

COLUMNS

PROGRAMMING PARADIGMS

by Michael Swaine

Michael returns to the topics of Java bashing, Java books, and JavaBeans.

93

C PROGRAMMING

by Al Stevens

In rethinking the relationship between C and C++, Al wonders if it is time to put the proverbial cart before the horse.

99

JAVA Q&A

by Cliff Berg

Cliff uses the JDK 1.1 multilanguage feature to create a multilingual currency calculator.

105

ALGORITHM ALLEY

by Will Schroeder and Tom Citriniti

Polygon decimation algorithms reduce the number of polygons in a mesh while maintaining a good approximation to the original data, leading to faster, more realistic 3-D graphics. Will and Tom show how these algorithms are used and create a decimated VRML file.

109

UNDOCUMENTED CORNER

by Robert R. Collins

Robert discusses the basics of in-circuit emulation, describing how it has become his primary software debugging tool.

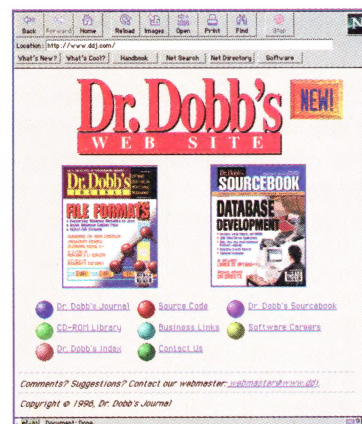
113

PROGRAMMER'S BOOKSHELF

by Gregory V. Wilson

This month, Greg examines Jonathan Rosenberg's *How Debuggers Work*, Mark Woodman's *Programming Language Choice*, and *The Media Equation*, by Byron Reeves and Clifford Nass.

117



FORUM

EDITORIAL

by Jonathan Erickson

6

LETTERS

by you

10

NEWS & VIEWS

by the DDJ staff

16

OF INTEREST

by Eugene Eric Kim

120

SWAINE'S FLAMES

by Michael Swaine

128

SOURCE CODE AVAILABILITY

As a service to our readers, all source code is available on a single disk and online. To order the disk, send \$14.95 (California residents add sales tax) to Dr. Dobb's Journal, 411 Borel Ave., San Mateo, CA 94402, call 415-655-4100 x5701, or use your credit card to order by fax, 415-358-9749. Specify issue number and date. Code is also available through the DDJ Forum on CompuServe (type GO DDJ), via anonymous FTP from site ftp.mv.com (192.80.84.3) in the /pub/ddj directory, on the World Wide Web at <http://www.ddj.com>, and through DDJ Online, a free service accessible via direct dial at 415-358-8857 (14.4 kbps, 8-N-1).

NEXT MONTH

August ushers in our annual C/C++ Programming issue.

Q: What does it take to build a superior client/server solution?
A: A SUPERIOR SOLUTION.

Java interface!
 New c-tree 6.7!
 Enhanced Servers!
 Check it out!!
www.faircom.com

START with the most advanced client-side SDK on the market: c-tree® Plus at \$895.

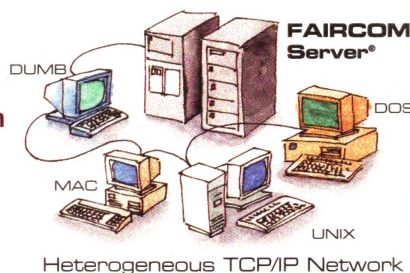
- Complete "C" Source code
- ROYALTY FREE (Client Side)
- Multiple supported protocols
- Fast, portable, reliable
- Powerful features like transaction processing
- Win95, NT, and Windows 3.1 ready

RESULT is economical, & deployable product that fits your needs.

- Portable
- Scalable
- Exceptional Performance
- Flexible
- Easy Server distribution
- Convenient OEM terms

ADD a strong, multi-platform, industrial-strength Server that supports.

- File mirroring
- Heterogeneous networking
- Automatic disaster recovery
- Multi-threaded design
- Best price/performance available: from \$445- \$3745



You can't find a better client SDK with these features!
Over sixteen years of proven reliability and performance.
No one else supports over 30 platforms in this price range!

c-tree Plus®

- Complete C Source
- Single/Multi User
- Client/Server (optional)
- Full ISAM functionality
- No Royalties
- Transaction Processing
- Fixed/Variable Length Records
- High Speed Data/Index Caching
- Batch Operations
- File Mirroring
- Multiple Contexts
- Unsurpassed Portability

FairCom Server®

- Client/Server Model
- Transaction Processing
- Requires <2MB RAM
- Online Backup
- Disaster Recovery
- Rollback - Forward
- Anti-Deadlock Resolution
- Client-side "C" Source
- Multi-threading
- Heterogeneous networking
- File Mirroring
- OEM/Source Available

CIRCLE NO. 84 ON READER SERVICE CARD
FOR YOUR NEXT PROJECT CALL FAIRCOM: YOU
CAN'T FIND A BETTER HETEROGENEOUS
CLIENT/SERVER SOLUTION!

Also inquire about these FairCom products:

d-tree™ r-tree® ODBC Driver

FAIRCOM CORPORATION®

WWWWeb Address: <http://www.faircom.com/>
800-234-8180

U.S.A. 4006 W. Broadway - Columbia, MO 65203-0100
 phone (573) 445-6833 fax (573) 445-9698

EUROPE Via Patriotti, 6-24021 Albino (BG) - ITALY
 phone (035) 773-464 fax (035) 773-806

JAPAN IKEDA Bldg. #3,4f-112-5, Komei-chou - Tsu-city, MIE 514 Japan
 phone (0592) 29-7504 fax (0592) 24-9723



Dr. Dobb's JOURNAL

SOFTWARE
 TOOLS FOR THE
 PROFESSIONAL
 PROGRAMMER

PUBLISHER Peter Westerman

EDITORIAL

EDITOR-IN-CHIEF Jonathan Erickson
 MANAGING EDITOR Kevin Carlson
 TECHNICAL EDITOR Tim Kientzle
 ASSOCIATE TECHNICAL EDITOR Eugene Eric Kim
 SENIOR PRODUCTION EDITOR Monica E. Berg
 ASSOCIATE MANAGING EDITOR Deirdre Blake
 EDITORIAL ASSISTANT Amy W. Wong
 ART DIRECTOR Bruce Gardner
 CONTRIBUTING EDITORS Al Stevens, Tom Genereaux,
 Scot Wingo, George Shepherd, Robert Collins, Cliff Berg,
 Steve Alexander, David Betz, Bruce Schneier, Mark
 Russinovich, Bryce Cogswell
 EDITOR-AT-LARGE Michael Swaine
 PRODUCTION MANAGER Denise Denis

CIRCULATION

DIRECTOR OF CIRCULATION Jerry Okabe
 GROUP CIRCULATION MANAGER Michael Poplaro

MARKETING/ADVERTISING

ADVERTISING DIRECTOR Charles P. Shively
 BUSINESS MANAGER Jennifer Teaford
 DIRECTOR OF MARKETING Valerie Dow
 DIRECTOR OF ELECTRONIC PUBLISHING Stan Barnes
 SALES AND MARKETING MANAGER Holly Vessichelli
 ACCOUNT MANAGERS see page 121

MILLER FREEMAN INC.

CHAIRMAN/CEO Marshall W. Freeman
 SENIOR VICE PRESIDENT/CFO Warren "Andy" Ambrose
 SENIOR VICE PRESIDENTS David Nussbaum,
 Darrell Denny, Donald A. Pazour, Wini D. Ragus
 SENIOR VICE PRESIDENT/SOFTWARE Regina Starr Ridley
 VICE PRESIDENT/PRODUCTION Andrew A. Mickus
 VICE PRESIDENT/CIRCULATION Jerry Okabe
 GROUP PUBLISHER Peter Hutchinson

DR. DOBB'S JOURNAL ISSN 1044-789X is published monthly by Miller Freeman, Inc., 600 Harrison Street, San Francisco, CA 94107; 415-905-2200. Periodicals Postage Paid at San Francisco and at additional mailing offices.

ARTICLE SUBMISSIONS: Send manuscript and disk (with article, listings, and letter to the editor) to *DDJ* Submissions, 411 Borel Ave., San Mateo, CA 94402-3522.

DDJ ON COMPUSERVE: Type GO DDJ.

DDJ ON THE INTERNET: editors@ddj.com or <http://www.ddj.com>

SUBSCRIPTION: \$34.95 for 1 year; \$69.90 for 2 years. International orders must be prepaid. Payment may be made via Mastercard, Visa, or American Express; or via U.S. funds drawn on a U.S. bank. Canada and Mexico: \$45.00 per year. All other foreign: \$70.00 per year.

FOR SUBSCRIPTION QUESTIONS, change of address, and orders, send e-mail to ddj@neodata.com or call toll-free **800-456-1215** (U.S. and Canada). For all other countries, call **303-678-0439**, or fax **303-661-1885**. Or write, *Dr. Dobb's Journal*, P.O. Box 56188, Boulder, CO 80322-6188.

BACK ISSUES may be purchased for \$9.00 per copy (includes shipping and handling). For issue availability, call **800-444-4881** in the U.S. and Canada. All other countries call **913-838-7500** or fax **913-841-2624**. Send e-mail to orders@mfi.com. Back issue orders must be prepaid. Please send payment to *Dr. Dobb's Journal*, 1601 W. 23rd Street, Suite 200, Lawrence, KS 66046-2700.

POSTMASTER: Send address changes to *Dr. Dobb's Journal*, P.O. Box 56188, Boulder, CO 80328-6188. ISSN 1044-789X. GST (Canada) # R124771239.

Canada Post International Publications Mail Product (Canadian Distribution) **Sales Agreement No. 0548677**.

FOREIGN NEWSSTAND DISTRIBUTOR: Worldwide Media Service Inc., 30 Montgomery St., Jersey City, NJ 07302; 212-332-7100.

Entire contents copyright © 1997 by Miller Freeman, Inc., unless otherwise noted on specific articles. All rights reserved.

Estimated print run 169,500.

Miller Freeman
 A United News & Media publication

ABP
 American Business Press

Printed in the
 USA

BPA

Dr. Dobb's Journal, July 1997

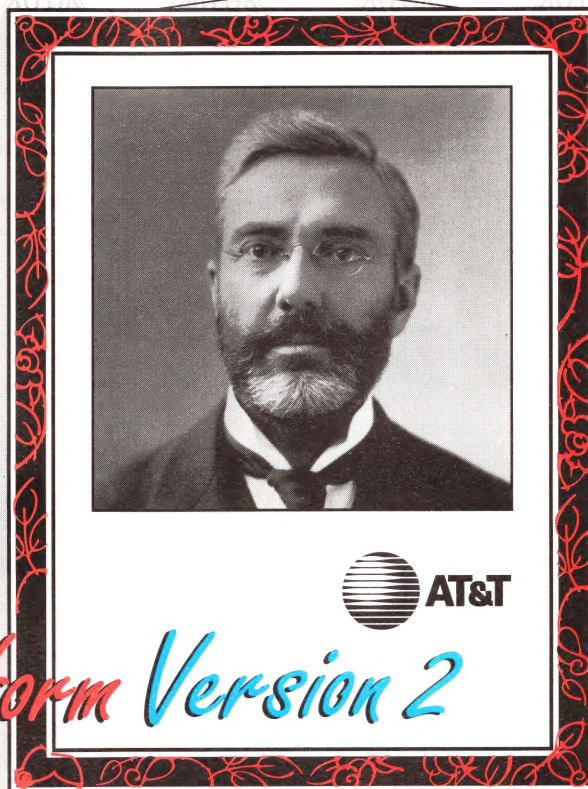
THE TECHNOLOGY
LEADER IS NOW THE
PRICE LEADER

\$495.00

tradition
meets
technology

WATSON™

Advanced Speech
Applications Platform Version 2



The AT&T WATSON Software Developers Kit is a resource for adding speech recognition and text-to-speech to your application. Packed with tools and sample code, this kit allows you to build powerful speech centric applications using industry standard Microsoft SAPI and TAPI.

Introducing Version 2.0 with Speaker Verification Improved Tools and Performance

AT&T Speech Recognition

Windows 95/NT V2.0 Technical Specification

- * Speaker Verification
 - SAPI and TAPI Compliant
- * Advanced Speech API, New Interfaces and C++ Class Encapsulations
- Sound Blaster™ compatible sound card and drivers required
- Subword Models, Whole World Digit Models
- * WATSON Grammar Studio™, Next Generation Compiler
- Background, Silence, and Filler/Garbage Models
- Barge-In speech begin/end notification (Barge-in hardware required)
- Uses 2 MB RAM, 10 MB Disk
- Pentium 75 MHz or higher CPU (uses < 50% CPU)
- * Improved Performance Larger Grammars. 100 Word Name Vocabulary or 200-300 Word Phrase Realtime Recognition, Natural Language Understanding
- Rejection of out-of-vocabulary words

* New in Version 2.0

Windows 95 and Windows NT are trademarks of Microsoft Corp.

© 1996 AT&T. All rights reserved.

AT&T FlexTalk™ Speech Synthesis

Windows 95/NT V2.0 Technical Specification

- Developed by AT&T Laboratories
- SAPI Compliant
- Male and Female Voices available
- 8 KHz and 11 KHz output
- Sound™ Blaster compatible sound card and drivers required
- Uses 2 MB RAM, 10 MB Disk
- Pentium 75 MHz or higher CPU (uses < 50% CPU)
- Context Sensitive Abbreviation Expansion
- Accurate pronunciation of most Proper Names
- Adjustable Vocal Tract Size, Speed, Volume, Pitch, etc.
- American English

To Order or for Information: 1-800-5WATSON

Visit our WEB site at: <http://www.att.com/aspg>



Random Readings

In the Spring 1997 issue of *The American Scholar*, historian and social observer Gertrude Himmelfarb bemoans the convergence of education, learning, and the Internet. In an essay entitled "Revolution in the Library," Himmelfarb, a professor emerita at the City University of New York and author of *The De-Moralization of Society: From Victorian Virtues to Modern Values*, notes that, while computers and the Internet are providing unparalleled access to information, they are also changing the very nature of learning and education. From Himmelfarb's perch, what the electronic revolution is fostering, teachers are allowing, and students are flocking to is a cut-and-paste approach to learning, where students glue together ideas and concepts without thinking things through.

Borrowing from the jargon of software development, the phenomena Himmelfarb describes could easily be dubbed "component-based learning." With this model, information—whether it be GUI controls or demographic surveys—is grabbed off the Internet and plugged into a framework of one kind or another. Among other advantages, component-based assembly requires less time and skill. Consequently, everything from automobiles to houses are assembled with premanufactured parts. Of course, applying this model to the assembly line is one thing; using it to mold young minds is quite another.

Himmelfarb does admit she is of two minds about this electronic revolution. On the one hand, she applauds the "democratization of access to knowledge," as does anyone who's accessed a major research library without leaving home. Himmelfarb is quick to note, however, that this should not be confused with "democratization of knowledge itself." She goes on to say that, unfortunately, "this is where the Internet...may be misleading and even pernicious, [because] in cyberspace, every source seems as authoritative as every other."

Therein lies the rub: The problem isn't with students having access to 11 billion words and 22 million web pages, but with their ability to weigh the value of the information they encounter. Lacking the intellectual tools for critically evaluating information, too many people—students and otherwise—end up assigning value to the quantity of amassed facts, rather than their quality.

This is neither a problem that technology causes, nor one it can solve. Although marketers would have you believe otherwise, fundamental knowledge and skills are required for component-based anything. Just as programmers need to know about data structures and algorithms before building apps using tools such as Visual Basic, PowerBuilder, or Delphi, students need to know how to think critically before they can effectively use the Internet as a research tool.

Fundamental skills in critical thinking eventually led Borge Nodland and John Ralston to publish a paper entitled "Indication of Anisotropy in Electromagnetic Propagation over Cosmological Distances" in *Physical Review Letters* (April 21, 1997). In their paper, Nodland and Ralston brought into question Albert Einstein's Theory of Relativity, which assumes a constant speed of light and a centerless, directionless universe.

When working on his Ph.D. at the University of Kansas, Nodland discovered that radio waves apparently rotate through space in a corkscrew pattern. In conjunction with Ralston (his advisor), Nodland (now at the University of Rochester) further found that radio waves traveling along the axis between the constellations Sextans and Aquila travel faster than those traveling at right angles to the axis. What this suggested is that the universe, like the Earth, may have its own axis—a notion contrary to Einstein's relativity theory. The research also brought into question Einstein's theories concerning the constant speed of light in a vacuum.

Both researchers emphasize that there are a number of other possible interpretations of the data, and that the effect of this universal axis is so small that their findings do not necessarily contradict previous findings. And you can bet fellow researchers will apply the same critical evaluations to Nodland and Ralston's theories that they applied to Einstein's.

Critical thinking skills must have been on hold when, according to the *New York Times* (April 28, 1997), the U.S. Patent Office granted patent 5,600,836, a "System and Method for Processing Date-Dependent Information which Spans One or Two Centuries"—the Year 2000 problem, in other words.

Rather than changing millions of lines of program code, the technique Harvey Alter "invented," and for which he was granted a patent, focuses on the data. "My idea is not to change the programs at all, but just subtract 20 or more years from all the dates in the data files," says Alter. "Then the relationship between the dates remains the same." All you really have to do, according to Alter, is turn back the system clock—and then pay him a royalty, of course.

A thumbs-up to Alter for creative thinking. Two thumbs-down to the Patent Office for the lack of critical thinking.



Jonathan Erickson
editor-in-chief

LICENSE TO SPEED



ImageGear: High Performance by Design™

Fastest Imaging Toolkit

Give your applications a License to Speed with ImageGear™ Pro Gold - the fastest image loading, displaying, image processing, compression and image handling available. No one wants to wait for images to display, and fast is never fast enough, so AccuSoft is committed to continuous optimization of algorithms and code to keep you ahead of your competition.

Over 50 raster formats

ImageGear now supports over 50 formats - more than any other product. With the unique AccuSoft Image Guarantee, it offers the most advanced support for TIFF, JPEG, GIF, MO:DCA, CALS, Photo CD, PNG, AVI and all the rest (see website for complete list). **Your image is important, so choose the only toolkit that guarantees it!**

Hundreds of features

ImageGear has the most comprehensive set of imaging features available: high speed display, printing, scanning, image processing, format conversion, compression, plus advanced features

like scale-to-gray, auto-deskew, noise removal, and all at unmatched speed. You get both high-level and low-level API for maximum ease of use and flexibility plus more of the features you need.

ImageGear Extensions

ImageGear is designed for easy expansion with optional Extensions that keep you ahead of the game. This allows you access to the latest technologies without waiting for new versions of the toolkit. You also get to choose only those components that you need. We offer industry-leading extensions for: annotations, FlashPix, DICOM, ABIC and other special formats, plus more on the way. The list will keep growing, so check the web site regularly.

Choose the Leader

We make your selection of an imaging toolkit easy by supporting the most formats, and by providing advanced features, faster performance, more options, and the best cross-platform support available. All of this has made AccuSoft the industry leader and your best source for imaging technology.

ImageGear Extensions

ART 2.0 Annotation Extension

The easiest way to add industry-leading annotations to your applications.

- Full support for the Wang spec for storing annotations in TIFF
- Available as DLL or ActiveX
- Read/write into TIFF or elsewhere
- Over 25 different objects
- Full hyper-linking
- Advanced redaction
- Easy internationalization & security
- High performance display
- Built-in toolbar & dialogs
- Fully automated user interaction
- Fully programmable user interaction

FlashPix Extension

The official and most complete FlashPix support.

- Supports complete set of advanced FlashPix features
- Includes over 30 special API functions
- Supports all FlashPix transformations
- Enables full integration between FlashPix and ImageGear
- Converts FlashPix to/from any other ImageGear format

DICOM Medical Imaging Extension

Advanced support for 8-16 bit grayscale medical imaging.

- Full read & write DICOM 3.0 support
- Easy and full access to all non-image data
- Advanced 16 bit image processing
- Full fidelity of images maintained
- High quality display
- including real-time Window Leveling
- Conversion of DICOM images to/from any other ImageGear format
- Full access to all ImageGear features
- Transfer syntax auto-detection

- No cost Evals available for all Extensions
- All toolkits available as DLL or ActiveX

Order Today
800-741-7494

AccuSoft®
High Performance Imaging™

©1997 AccuSoft Corporation. All Rights Reserved. AccuSoft is a registered trademark of AccuSoft Corporation. High Performance Imaging, ImageGear, the ImageGear logo and "License to Speed", are trademarks of AccuSoft Corporation. All company and brand names are trademarks or registered trademarks of their respective owners. License required for formats using LZW compression.

AccuSoft Corporation Two Westborough Business Park Westborough, MA 01581 Tel (508) 898-2770 Fax (508) 898-9662

CIRCLE NO. 65 ON READER SERVICE CARD.

Visit our Website
www.accusoft.com
Special Offers, Bundles, Demos and more!

Tool One:

Codewright users win



An Editor whose unique blending of standard features and innovation lowers the hurdles in your way.

A satisfaction guarantee like no other in the industry.

Qualifications: (continued)

- Brief emulation
- API Assistant
- Outline Symbols
- Multi-file replace
- IDE integration
- C/C++ support
- HTML/Java support
- Tabbed dialogs
- Advanced Diff

Free and unlimited telephone and Web support from a team of world-class professionals.

from

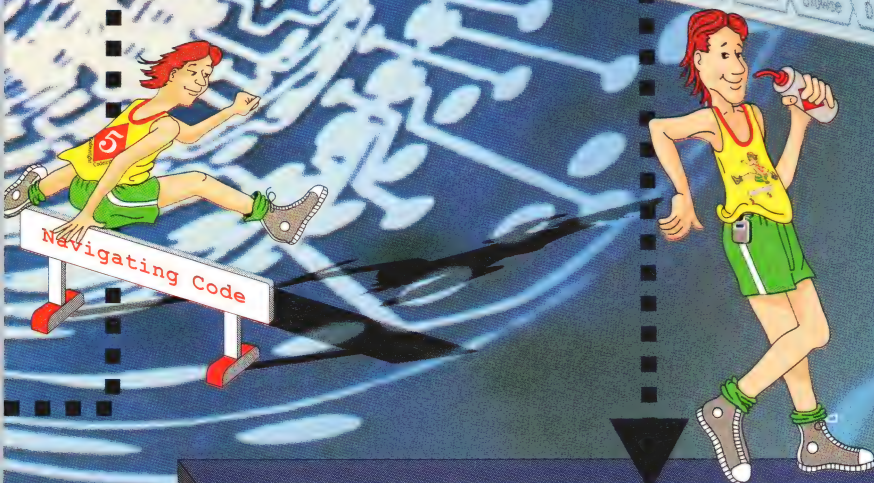
premia®
Premium Quality Software

the productivity race.

A "live copy" evaluation program and a trial version you can download, so that you can try before you buy.

premia.com

Codewright™ v5.0
Professional
Programmer's Editor



It's people like **YOU**, looking for ways to work better and easier.

Call for an evaluation now: 1-888-4-PREMIA
4 7 7 - 3 6 4 2

\$269

Single user license for
Windows, Windows 95, and NT.
Volume discounts and site licenses available.

Premia Corporation
1075 NW Murray Blvd., Suite 268
Portland, Oregon 97229 USA
Phone: +1-503-641-6000
Fax: +1-503-641-6001
sales@premia.com
www.premia.com

**C++Builder**

Dear *DDJ*,

As a former Borlander and a member of the Delphi 1.0 development team, I would like to correct a misstatement Al Stevens made in his "C Programming" column (*DDJ*, May 1997).

Zack Urlocker was indeed the project manager for Delphi, but he was not the main developer. That honor goes to Anders Hejlsberg, who was the original author of Turbo Pascal and the architect and lead developer of releases 1 and 2 of Delphi, before he defected to Microsoft a few months ago.

Incidentally, Delphi never had the code name Ivory, although a wide variety of code names were used for the beta versions distributed under NDAs (so that Borland could detect the source of any leaks).

I've used both languages extensively, and I think if Al were to overcome his prejudice and take the plunge, he would find that Object Pascal is at least as good a language as C++ for application development, and arguably a better one.

Mike Harrison

San Jose, California

mikeh@slip.net

Al responds: Mike, thanks for straightening me out on Anders. Overcome my prejudice? I didn't think I had one. I looked over what I wrote and couldn't find any Pascal bashing there. I apologize if it sounded like that. I disagree with your assessment of the relative merits of the two languages, though; but I think of it more as a preference than a prejudice. The industry seems to agree with me. But I think it's great that we now have the neat Delphi platform for both languages so everybody can be happy. Again, many thanks for your comments.

Dear *DDJ*,

I read Al Stevens' "C Programming" column about C++Builder (*DDJ*, May 1997). I, too, tried the beta version and fell in love with it. I purchased it as soon as it was available. It will certainly be a hit. It's

nice to have the advanced feature of C++ combined with a wonderful development environment for once.

One thing about Al's column that surprised me was his disinterest in Delphi. After five years of C++ development, I was not interested in Delphi either, at first. Until a recent project at IBM, I was given a choice of any 16-bit tool I wanted, except VB 4.0, which was too slow. After trying to go back to VC++ 1.52, I was so repulsed by the lack of features that I was accustomed to from VC++ 4.2 that I tried my only alternative, Delphi. I was turned off by having to use Pascal, but there is no comparison to VC++ 1.52 in terms of features and ease of use. The thing that changed my opinion was the almost instantaneous compile times. The compile times, in conjunction with support for object-oriented features and low-level support, make it such a great product. For a small- to mid-sized project, the thing takes only 3–5 seconds to build and run! The only thing I really miss dearly is support for templates. Maybe it's not possible because Pascal doesn't have a precompiler. At least now that C++Builder is here, I can use templates again. With the new incremental compiler and linker, it at least seems fast sometimes. I love the advanced features of C++, but if you need speed, you can't beat Delphi.

John Bloch

Bloch-head@worldnet.att.net

Dear *DDJ*,

As someone who has been desperately awaiting the release of a quality C++ RAD tool, I was interested to hear Al Stevens' opinions about C++ Builder in his "C Programming" column (*DDJ*, May 1997). Like Al, I've been pretty impressed, but also like him, I think I received a beta. I find this unfortunate in that I was led to believe that I was purchasing a release version.

As a data-control ISV, we find the namespace problem pretty ugly. Suddenly, several of our most-commonly used functions don't ever get called (things like *Insert* and *Update*). Even after coming up with a workaround, we still have to be careful about where we place our header files! I think the cutest little bug, though (the one that almost cost my monitor its already-refurbished life), is that if you build a project while the code window is minimized, one warning or error will completely snafu the machine. Apparently, the code window gets normal-sized and is put in stay-on-top mode when the error or warning is found. This is nice for viewing the problem, but not so nice for clicking on

the OK button in the modal Build Project dialog.

I take slight issue with one of Al's objections to the product, however—I really like having the VCL components accessed by pointers. To me, it feels like I'm "really programming." It's a psychological lift to go from a RAD environment in which pointers, like idle time processing, are considered the devil's work, to one in which pointers are bandied about like a beach ball at an outdoor concert. We're really in C++! Besides, no one I know makes pointer errors ;>). BTW, my car also has a manual transmission.

I'll bet Borland catches a lot of heat, more so from the fact that it released such a promising product with a few bugs and poor docs than from just putting out a bad tool. Since I'm very excited about this product and want to make it one of my primary tools, I find the bugs much more annoying than bugs in a lesser-used and not as cool product. I suspect that I am not alone.

Ken White

ken@smithware.com

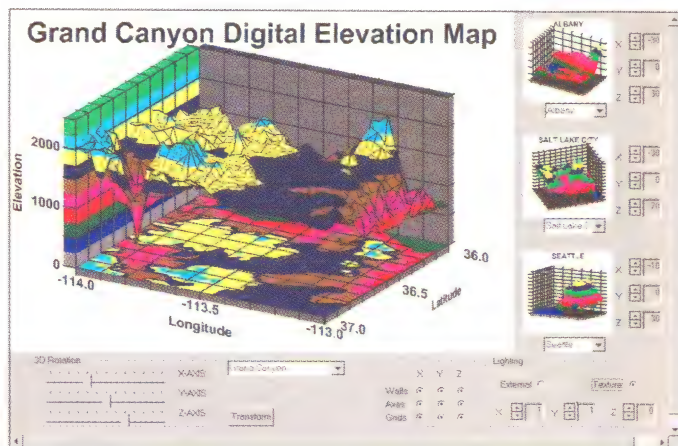
Dear *DDJ*,

Al Stevens' "C Programming" column about Borland C++Builder (*DDJ*, May 1997) was great. Perhaps I can add a couple points that explain why some things in C++Builder operate as they do.

All the VCL classes have to be treated as pointers because the underlying VCL (Object Pascal) code is designed that way. In Delphi, the decision was made to turn *all* references to classes into pointers. In Delphi, it is simply impossible to create a class without creating it on the heap. This was done for performance- and memory-allocation-related reasons, and also because it let us simplify notation. Since all Delphi classes are allocated on the heap, for instance, we could treat them all with the same syntax. And since pointer syntax is confusing, why not let users reference those pointers with dot notation, as if they were static classes?

In other words, since all classes had to be pointers, we had the luxury of letting you act as though they were static references, since the compiler knew it had to be a pointer, even if you didn't use pointer notation. Furthermore, most allocations and deallocations for these classes are handled automatically by the VCL.

Unfortunately, there was no simple way to give the same syntax to C++ programmers, since the rules of ANSI C++ wouldn't let us change the meaning of pointer syntax globally, the way we did in Object Pascal. Of course, allocation and deallocation for classes is still usually handled automatically, as it is in Delphi. However, in



Four separate 3D surface charts with contour texturing are combined with controls in a CFormView derived class.

3D with No Program Limits

Number of data points in a graph - **NO LIMITS**
 Number of chart objects in a graph - **NO LIMITS**
 Number of axes in a graph - **NO LIMITS**
 Number of text objects in a graph - **NO LIMITS**
 Number of 3D graphs in a view - **NO LIMITS**
 Number of legends in a view - **NO LIMITS**

3D++ and OpenGL™

The 3D graphics engine used by the 3D++ class library is OpenGL, the 3D graphics library developed Silicon Graphics, Inc. OpenGL is the foundation for many of the most advanced 3D imaging systems in the world. Originally developed for use in the Silicon Graphics line of graphics workstations, Microsoft licensed a version of OpenGL for use with Windows NT and Windows 95. OpenGL is a Microsoft "redistributable" and can be included in third party application programs without royalties. Many of the newer high-end PC graphics boards include hardware accelerators for OpenGL. Quinn-Curtis analyzed all of the 3D graphics engines currently available for PCs and decided that only OpenGL has the horsepower necessary to propel applications into the 21st century.

Ordering Information

3D++ WIN-M3D-300 \$500

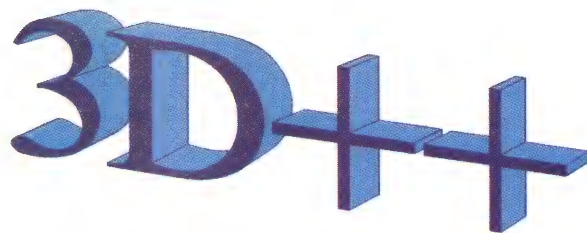
Free Demo Disk and Catalog

Call, FAX or write for a free catalog and demo disk. Product information and demos can also be downloaded from our WWW, FTP and BBS sites.

Order Now !

Tel. 617/449-6155 FAX 617/449-6109

30-Day Money-Back Guarantee !



A C++ Class Library for 3D Visualization Under Windows NT™ and Windows® 95

The 3D++™ class library is written in C++ from the ground up using Visual C++™. 3D graphs can be placed in any MFC view, occupying the full view or any portion of a view. 3D graphs can also be combined with controls in dialogs based on CDialog and views based on CFormView. The library is implemented as a DLL which adds the 3D classes directly to your application program.

The 3D++ class library is for use with Visual C++ 4.0 and higher and MFC. It includes a comprehensive manual documenting the 300+ member functions. On-line help and 20 example programs are also included.

3D++ Features

Axes

Axes with Tick Marks
 Log axes
 Walls
 Grids
 Arbitrary XY Intercepts
 3D Rotated Axis Labels
 String Labels
 Manual and Auto-Axes Routines
 Multiple Axes, Multiple Scalings

Chart Objects

Simple Chart Objects

Line Ribbon
 Bar Area Fill
 Range Fill Scatter Symbol
 Scatter Text Plane

Group Chart Objects

Stacked Bar Stacked Ribbon
 Stacked Fill Group Bar

Compound Chart Objects

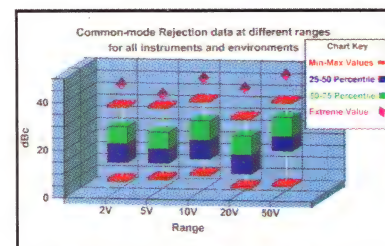
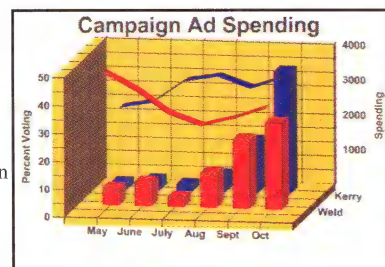
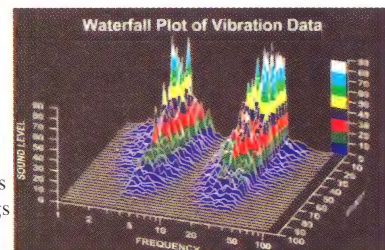
Open-High-Low-Close
 Box and Whisker

Polysurface Chart Objects

3D Polysurface
 2D (Projection) Polysurface
 Contour Lines and Surfaces

Miscellaneous

Lighting Serialization
 Legends
 3D TrueType Text



Quinn-Curtis, Inc.

35 Highland Circle
 Needham MA 02194 USA

WWW <http://www.quinn-curtis.com> FTP <ftp.webcom.com> in /pub/quinn/demofiles BBS 617/449-4783

3D++ is a trademark of Quinn-Curtis, Inc. All other trademarks are the property of their respective manufacturers.

(continued from page 10)

BCB, you must use pointer notation when referencing VCL classes.

The keyword `__published` refers to properties that will appear in the Object Inspector. If you publish a property, then it will appear in the Object Inspector. There was no existing keyword in C++ that had the same meaning, so we had to make up our own. Somewhat more technically, `__published` tells the compiler to give properties all the RTTI they need to appear in the Object Inspector.

Once again, thanks for your great article.

Charles Calvert
Borland International
ccalvert@corp.borland.com

Dear DDJ,

I read Steve Barrett's letter in the May 1997 DDJ, and would like to offer some information on the subject of the computers used in previous NASA missions. I have found a good nontechnical book on the subject: *Computers in Space, Journeys with NASA*, by James E. Tomayko (Alpha Books, 1994, ISBN 1-56761-463-9).

From this source, the systems that we would recognize today as computers were all kept on the ground. The systems used in the capsules seem to have been special-purpose controllers built to meet the needs of the spacecraft sensors and controls. The Acceptance Checkout Equipment (ACE) used two CDC-168s while the launch systems used RCA-110As.

Mission Control (by the time of the Apollo missions) ran on IBM 360s with a real-time modified version of OS/360 called RTOS/360. The Fortran code from the Gemini days moved from IBM 7094s more or less intact.

The Apollo guidance computer that flew in the spacecraft was designed by the Instrumentation Laboratory at MIT, apparently based on work done by Dr. Charles Draper for the Polaris missile-guidance system. This guidance system ran pre-computed programs designed before the mission to operate the spacecraft sensors and controls. I can't type in all of the information in the book, so let me summarize it as best as I can.

The Apollo computer had 50,000 NOR gates in integrated circuits for the ALU. By the time of the Moon landings, the chips were "ancient" technology. The size was 2x1 feet by a half-foot. The word size was 16 bits, and the machine was a fixed-point processor. The memory started small, about 4000 words in permanent storage and 256 words in magnetic core memory. By the time of the Moon landings, this grew to 36,000 words of nonerasable and 2000 words of erasable memory. Most of the programs were assembled by hand in a "core rope;" permanently magnetized rings were lined up and the wires of the data bus were "woven" through them to record the data patterns. Magnetic core memory was preferred for its invulnerability to radiation damage. This hand-woven memory was a production nightmare. Secondary storage memory was a magnetic tape system. Tape recorders are still used in spacecraft designs like the Galileo. The Lunar lander may have been the pinnacle of Apollo guidance computers with the TRW MARCO 4418 (for "MAN Rated Computer"), it weighed "only" 32 pounds and required 90 watts of power.

The book is very interesting, and it lists for \$20. (I have no connection to Alpha Books or the author.) Anyway, Al Steven's column is my favorite part of *Dr. Dobb's Journal* and I hope you find this info entertaining.

Robert Brauer
Robert_N_Brauer on CompuServe

DDJ

DDJ welcomes your comments and suggestions. Send letters to DDJ, 411 Borel Avenue, Suite 100, San Mateo, CA 94402-3522. We can also be reached via editors@DDJ.com, 76704.50@compuserve.com, and by fax at 415-358-9749. Please state your name and address. DDJ reserves the right to edit letters for length and content.

Real-time Windows NT®:
it just takes imagination.

real time for the mainstream

highly deterministic performance

run multiple Windows apps concurrently

use Microsoft Visual C++

preemptive, multi-threaded

50 microsecond typical latency

message passing ipc

irq services

already proven in commercial apps

ActiveX™ communications

real time that really rocks

contact us today for the whole
Hyperkernel™ story

IMAGINATION
SYSTEMS INCORPORATED

A wholly owned subsidiary of Nematron Corporation

<http://www.imagination.com>

1-800-331-2565
1-757-497-8200

© 1997 Nematron Corporation.
Hyperkernel and Imagination Systems are trademarks, and the Nematron logo is a registered trademark of Nematron Corporation. Windows and Windows NT are registered trademarks and Microsoft Visual C++ and ActiveX are trademarks of Microsoft Corporation.

CIRCLE NO. 353 ON READER SERVICE CARD

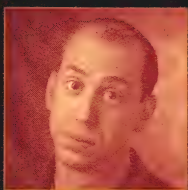
Develop Java apps



edit



compile



link



load

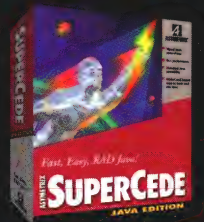


run

while you're still young.

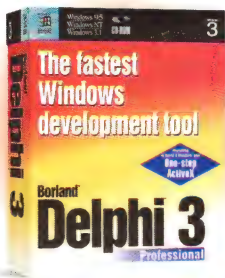


Now thanks to SuperCede's interactive development environment, it won't take ages to develop applications. Because with our Flash Compiler™ technology, you can make changes while your application is running—helping you work five times faster. Only SuperCede lets you create high-performance native executables and platform-independent applets. And offers seamless C++/Java interoperability. Plus with our new editions, you get dozens of Java and ActiveX controls and total database support. For the whole story, download SuperCede at www.supercede.com and try it free for 15 days. Or call 1-800-448-6543 for your nearest retailer. After all, you're not getting any younger.



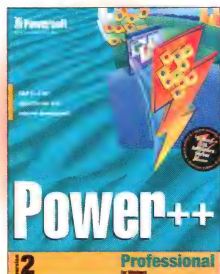
800.448.6543 www.supercede.com

CIRCLE NO. 152 ON READER SERVICE CARD



\$269

Paradise No. BNUD88-FT
Borland Delphi 3 Pro Upg.
by Borland International



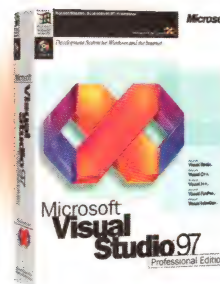
\$489

Paradise No. PSPUA88-FT
Power++ Pro Comp. Upg. 2.0
by Sybase/Powersoft



\$9.99*

Paradise No. MSOBW88-FT
Microsoft Visual J++ 1.1 Professional Version Upg.
by Microsoft Corporation



\$359.99*

Paradise No. MSOCF88-FT
Microsoft Visual Studio 97 Professional Version Upg.
by Microsoft Corporation

* Price after manufacturer's direct rebate.

In Stock Upgrades. Order Today!

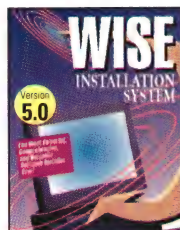
ad•van•tage

(əd van' tij) *n.* superiority, benefit or gain.



WISE Installation by Great Lakes Business Solutions

Creates professional installations for Windows, Windows NT, and Windows 95 all with a single installation script. The WISE Installation System includes both 16- and 32-bit versions of the integrated development environment. WISE includes all of the standard installation features plus: custom dialog/graphic editors; CD-ROM/Network installations; multi-lingual support; and full Windows 95 install requirements.



Paradise No. GLISV31-FT
\$169

SuccessWare Data! Engine Suite by SuccessWare

You're a seasoned professional. You've learned the hard way that what's free in the box is generally worth every penny. You demand more from the tools you choose. The SuccessWare Database Engine Suite offers plug-n-play compatibility, absolutely tiny (250k) deployment and rock-solid multi-user access to FoxPro & Clipper index and memo files from Microsoft and Borland development environments.



Paradise No. SWDES31-FT
\$295

WinDK™ & SoftICE™ Device Driver Development by BlueWater Systems, Inc. and NuMega Technologies

The best NT® Device Driver Library bundled with the best NT debugger. The WinDK Device Driver Development Kit is the most efficient way to develop high speed, high quality device drivers for Windows NT and WDM. SoftICE for Windows NT is the industry-standard debugger. Create bulletproof drivers and save yourself weeks of work.

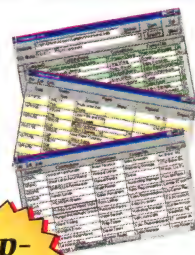


Save \$274!

Paradise No. BSWB88-FT
\$1,069

Ultimate Grid 3.0 for MFC by Dundas Software

Includes 100% MFC sourcecode. Flexible and efficient design integrates with any type of data store, including SQL, DAO, ODBC, and others. If you develop in MFC, drop your ActiveX control and let Ultimate Grid for MFC save you weeks of development time.



Top-Seller

Paradise No. DSUGA31-FT
\$375

Multi-Edit with WebLair by American Cybernetics

Now the source editor known world-wide for power and flexibility includes WebLair's complete Web development technology. Supports HTML, Java, JavaScript, Perl, and VRML. Write JavaScript within an HTML file, with full language and syntax support for each. Features Editable Templates and Common Code Management—edits to HTML text can be instantly reflected across multiple files, and more!



Paradise No. AMWLB31-FT
\$299

WinSCSI-32 by Sequoia Advanced Technologies, Inc.

Full 32-bit SCSI command link library with over 150 functions designed to make your SCSI application programming tasks easier. Complete with extensive sample source code and electronic online documentation, WinSCSI-32 is the ideal choice for all your SCSI programming needs. For Windows 95 and Windows NT. Royalty-free. Visit www.pparadise.com for more product information!



Paradise No. SEWS288-FT
\$175

c-tree Plus® by FairCom

DOS • WINDOWS • NT • UNIX • OS/2 • SUN • RS6000 • HP9000 • MAC • QNX • BANYAN • SCO. This well known, highly portable data management package has become established as the tool of choice for commercial development. Offering unprecedented data control, choose from direct low level access, ISAM level, or ODBC access with the FairCom Server. Single User, Multi User, or optional Client/Server, ANSI Standard. Full Source. No Royalties.



Full Win 95 support!

Paradise No. FACTP31-FT
\$785

Spread v2.5 by FarPoint Technologies

A complete spreadsheet control for most environments that support a VBX, OLE control, or DLL. Use as a spreadsheet to obtain variable lines of data or display tables of information. Data-aware—connect to databases with the Access Engine and ODBC. Includes over 250 properties and our improved Spread Designer. Formulas, sorting, and full-print support too.

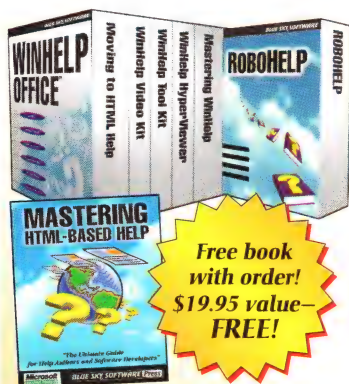


Paradise No. FPSOX31-FT
\$229

Order online 24 hours a day, every day at www.pparadise.com

Programmer's Paradise®

the developer's definitive source for software!



**Free book
with order!
\$19.95 value—
FREE!**

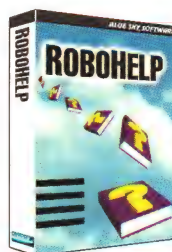
WinHelp Office® 4

by Blue Sky® Software

"The Complete Help Authoring Solution"

WinHelp Office 4 includes everything you need to quickly and easily produce professional Help systems for Windows 3.1, Windows 95, and Windows NT (4.0 & 3.51), Microsoft HTML Help, Netscape NetHelp, printed documentation and intranet/Internet Websites—all from a single source! WinHelp Office is power-packed with the award-winning RoboHELP, Mastering WinHelp, WinHelp Video Kit, WinHelp Tool Kit, WinHelp HyperViewer, Moving to HTML Help and more!

WinHelp Office includes RoboHELP.



RoboHELP

Paradise No.
BSRWD88-FT

\$479

WinHelp Office

Paradise No.
BSW9788-FT

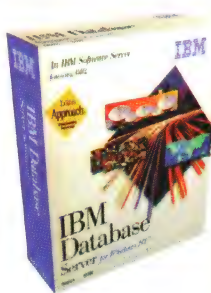
\$679

Slay those BLOBs with DB2® V2.1!

by IBM

DB2 universal database for Windows NT® lets you master multiple data types better than other DBMSs. Can your NT database search and retrieve 2 gigabyte BLOBs based on their color, shape or pattern? Can your NT database master video, audio, images, fingerprints or whatever data type you throw at it? Can your NT database sport the "Designed for Microsoft® BackOffice™" logo? DB2 can. DB2 can do a lot more than that, too. Java support, for instance. And triggers that let you enforce business rules, validate input and even alert users about changes to data. Plus a really slick query optimizer that selects the fastest access path to the data. And we gave it a multi-threaded architecture to leverage both the operating system's and the hardware platform's ability to multi-task. Oh, and did we mention that this engine really screams? Or that PC Magazine said DB2 was "state-of-the-art" and offered "strong performance on almost every front"?—3/26/96. No kidding.

So Web-enable to your heart's content. Get creative. Check out the DB2 universal database for Windows NT today. Windows and Microsoft are registered trademarks of Microsoft Corporation.



Product Name

Paradise No.

IBM DB2 for Win NT

Single-User

53H7473-FT

\$339

IBM DB2 for Win NT

Multi-User

53H7474-FT

\$1359

Codewright by Premia

You'll work faster and easier when you use the right programmer's editor, Codewright. Now you'll browse code faster with Outline Symbols. It gathers information about your code in the background. You'll juggle changes with ease, with Difference Editing. It lets you selectively combine the changes from two revisions. The new Bookmarks Window lets you view bookmarks by name and by file. Now with synchronizing technology for Delphi and Visual C++ IDEs. With the help of the API Assistant, making complex function calls is as simple as filling in a form. **Intro price.**



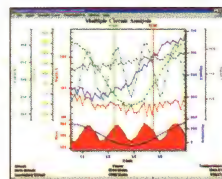
**New
Version
5.0!**

Paradise No.
PCCW588-FT

\$189

ProEssentials™ by GigaSoft®

16- and 32-bit DLL, OCX, VCL, and VBX interfaces providing charting functionality with consistent visual quality, real-world practicality, and overall professional appeal. Suited for engineering, financial, data-acquisition, and information system development. Be sure to find the best tool for your important development needs. Download a **demo** [500K] or get a **fully functional evaluation edition** [3Meg] at www.pparadise/publishers/gigasoft.



Paradise No.
GIPRB31-FT

\$299

KEDIT for Windows 1.5

by Mansfield
Software Group

Includes both 32-bit Windows 95/Windows NT and 16-bit Windows 3.1 modules. KEDIT is a powerful general purpose text editor with redefineable keys, undo/redo, selective line editing, regular expression support, enhanced syntax coloring, column oriented editing, file locking, a macro debugger, an IBM XEDIT-compatible command set, prefix area support, and more! The macro language is a subset of REXX. Also available in DOS and OS/2 text mode versions.



Paradise No.
MANKE31-FT

\$139

List Pro

by FarPoint Technologies

A set of customizable List Box and Combo Box controls. Create header groups and sub header groups and/or multiple columns to create a very unique list control. Merge cells in rows with same text to make display more user friendly. Bind to supported databases using Microsoft Jet Database Engine or Microsoft Access ODBC. Virtual memory manager allows connection to any size database. Supports single or multi-select lists, embedded icons and bitmaps in columns and rows plus drag-and drop capabilities.



Paradise No.
FALP131-FT

\$189

Order online!

These products and
many more are featured at
www.pparadise.com

Gain the advantage over your competition when you take advantage of the many opportunities presented by the Programmer's Paradise catalog! Programmer's Paradise features the best prices on outstanding development tools from today's top manufacturers. Plus, with Programmer's Paradise you'll also get exceptional service—guaranteed!

Check out the products appearing on these pages. They're just a small sampling of what you'll find in the Programmer's Paradise catalog.

To place your order, call today and take advantage of our convenient, quick 2-day delivery to any location in the continental U.S.!

A Personal Guarantee!

"If you're not happy, we're not happy! You have my personal guarantee that we'll provide you with outstanding products, support and service."

—Roger Paradis

President & CEO
Programmer's Paradise, Inc.

ORDER NOW
or call for your free catalog!
1-800-445-7899

Call for shipping charges/return policy.
Prices subject to change without notice.



CIRCLE NO. 128 ON READER SERVICE CARD

To Order Call 800-445-7899



or FAX 908-389-9227

Speak Up

Whether or not you think speech is the next great user interface, speech-enabled applications are coming. Even the exhibitors at Advanced Speech Applications & Technologies (ASAT) '97 were surprised by the number of developers moving from booth to booth, looking for the best tools to speech-enable applications.

The highlight of the exposition was Dragon Systems' natural speech recognition. Traditional speech-recognition applications require users to speak discretely, pausing between every word. Dragon Systems' latest offering recognizes continuous speech, allowing users to speak naturally. The system requires 18 minutes of training and is speaker dependent.

While Dragon Systems' new system might make dictation more attractive for text input than typing, it doesn't explain the rapidly growing interest in speech-enabling personal computer applications. Part of the appeal seemed to stem from the use of Microsoft's OS as a development platform. Several exhibitors demonstrated telephony applications using Microsoft's TAPI. Others showed how more nontraditional applications (such as a car diagnostics program that you could interact with while working underneath the vehicle) could be written in scripting languages like Visual Basic.

Mostly, however, developers seemed to express interest in speech recognition because it's there. Everyone is buying new computers these days that have more than the minimum hardware required to run speech applications, and the plethora of offerings exhibited at this conference showed that speech can be integrated into applications relatively easily and quickly.

—Eugene Eric Kim

From the "For What It's Worth" Department

In a recent report, the market-research firm Strategic Focus projected the Internet/intranet software market will grow from \$2.6 billion in 1996 to over \$7.2 billion in the year 2001 at a compounded annual growth rate of 23 percent. Among the 15 software market segments analyzed, the fastest growing segment was web-server software, followed by electronic commerce and web-application development tools. In the web-server

software market, Strategic Focus found that Netscape currently holds the highest market share at 11 percent in 1996 (measured by the revenue dollars generated), followed by Microsoft (8 percent), Novell (8 percent), and IBM/Lotus (7 percent).

—Jonathan Erickson

Help for the Visually Impaired

Researchers at Texas Instruments have come to the aid of visually impaired computer users by developing a display that creates renewable, raised dots on computer monitors and other screens. Marvin Cowens, Alan Gilkes, and Larry Taylor have been awarded a patent for the system which can be used to generate Braille "characters," thereby enabling the visually impaired to read the display. Current generation Braille displays rely on expensive and cumbersome mechanical pegs and moving parts.

According to the patent, the TI invention "consists of a matrix of small cavities, each containing a positive and negative electrode, and filled with...polar organic gel responsive to electric fields." An elastomeric film is then stretched over the matrix and voltage is applied to individual cavities, causing the gel to expand and create a raised dot on the film. The dots are the standard Braille size of 1.5 millimeters, but can be adjusted.

—Jonathan Erickson

Yet Another Intel/AMD Spat Settled

Intel and AMD have announced they've settled yet another trademark dispute, this time over AMD's use of the "MMX" tag in marketing its AMD-K6 chip. Intel claimed that MMX is a brand name, describing Intel-developed graphics, audio, video, and communications technology. Under terms of the agreement, AMD can use the term, but must acknowledge Intel's ownership of the trademark.

—Jonathan Erickson

One-Touch Divorce

California's Sacramento County Courthouse is the latest legal venue to install QuickCourt, an interactive information kiosk made by North Communications ([http://](http://www.infonorth.com/)

www.infonorth.com/). The system offers court information in English or Spanish and, with the touch of a button, can produce forms for users filing on family law matters—such as dissolving or nullifying a marriage—or small claims. Comparable systems are already in operation in Arizona and Utah. In addition to QuickCourt, North Communications produces other touch-screen systems for aiding the public in dealing with the long arm of the law, such as the AutoTouch driver-testing system, a Traffic School alternative for some Los Angeles County traffic violators, and AutoClerk, an ATM-like kiosk that accepts payments for traffic citations.

—Deirdre Blake

Silicon Valley Civil Suit Gets Criminal

What was a civil battle between California-based Avant! and Cadence Design Systems recently moved into the criminal arena, when the Santa Clara County District Attorney's Office filed felony charges of conspiracy and theft of trade secrets against several Avant! executives. The two companies have been sparring in civil courts since 1991, when several Cadence employees left the company to form Avant! Corp. Both companies produce chip-design software. The D.A. claims criminal charges are the result of an investigation by its High Technology Crimes Unit, which has reportedly found evidence that the defendants used stolen Cadence technology to develop competing products. Interestingly, in regard to the civil suit, Avant! claims that the U.S. District Court issued a ruling favorable to Avant! in March. The federal judge in that case found no use of misappropriated code in Avant!'s wares and rejected Cadence's motion to enjoin the sale and support of Avant!'s products. Avant! President Gerald Hsu denies all charges and stated that he "believe[s] that the D.A.'s office has been misled and misinformed by other parties in this matter." For Cadence's part, company president Joe Costello stated, "Cadence, the victim of these charged crimes, is pleased that the D.A. has decided to prosecute Avant!," adding that Cadence intends to continue pursuing the civil action against Avant!, as well.

—Deirdre Blake



Visual Basic™
Visual C++™
Visual InterDev™
Visual J++™
Visual FoxPro™

With Microsoft
Visual InterDev,
*your Web site becomes an
interactive bridge between
business and customer.*

*(And who knows
what'll
happen
when those two
get together.)*

Until now, building interactive Web sites that provide dynamic content to your customers has been an arduous task. Introducing Microsoft® Visual InterDev™, the first complete development tool designed to help you build database-driven Web applications quickly and easily. Visual InterDev has been designed for teams of developers and content providers to work on Web applications simultaneously. It includes programming and editing tools, database management, content creation, site management,

and multimedia tools. You could say it's everything you need to develop, publish, and manage a truly interactive and dynamic Web site. For a self-paced tutorial, get *Mastering Web Development Using Visual InterDev*. Visual InterDev is included in Microsoft Visual Studio 97, the comprehensive suite of award-winning development tools. To get started, visit us on the Web today.

Microsoft®

Where do you want to go today?™

www.microsoft.com/VisualTools/

©1997 Microsoft Corporation. All rights reserved. Microsoft, Visual Basic, Visual C++, and Where do you want to go today? are registered trademarks and Visual FoxPro, Visual InterDev, Visual J++, and the Visual Studio logo are trademarks of Microsoft Corporation. To order any of the Microsoft Visual Tools, or to receive a reseller referral, in the United States or Canada, please call (800) 621-7930, Monday through Friday (7:00 a.m. to midnight, Eastern time), Saturday (8:00 a.m. to 8:30 p.m.), and Sunday (8:00 a.m. to 7:00 p.m.). Outside the U.S. and Canada, please contact your local Microsoft subsidiary. TT/TDD available at (800) 892-5234.

Implicit Surfaces and Real-Time Graphics

Introducing the Silly Space modeling application

Kyle Lussier

Implicit surfaces are generally defined by an equation model that provides a logical basis for creating a closed surface in space. Implicit surfaces are useful in that they can be used in areas such as molecular modeling, animation, geophysical systems, and weather analysis.

Implicit surfaces are at the heart of Silly Space, the modeling application I'll describe in this article. Silly Space (available at http://www.inside.ch/silicon_softworks/ or http://inside.net/silicon_softworks/) lets you manipulate implicit surfaces in a real-time, click-and-drag 3-D environment under Windows 95/NT. Silly Space works by integrating an equation model, 3-D user interface, and surface tessellator into one working body. These components fit tightly together to provide an easy-to-use, robust modeling application.

Figure 1 is a function that describes the implicit surface of a two-component blob (or "meta-ball," as it is more commonly referred to). Figure 2 is the two-component blob modeled in Silly Space. By plugging in arbitrary values for x , y , and z in Figure 1, you can test where the function changes sign. It is at this point that a surface needs to be determined for visualization.

Of course, there are numerous methods for building a visualization of this surface. In raytracing, an exact evaluation of the surface is done for a displayed pixel. The surface can also be decomposed with voxels (volume pixels) to show the variations

in density of the space given by the equation. In Silly Space, I decompose the surface into a set of triangles using a polygon tessellator.

Multi-Geometry Surface Model

Implicit surfaces can be generated in many different ways, the most common being to design an equation that models the desired surface, then visualize or model it. However, this is cumbersome, time-consuming, and error-prone. In the interest of simplicity and flexibility, I used a multi-geometry surface model (MGSM)—a component-based implicit surface model. Figure 3 presents the general form of the MGSM model.

The goal of this model is to allow interactive user selection of various components and shapes and to connect this with the user interface. The quantities in the model are:

- *Threshold*, which specifies how tight the surface is in relation to the components of the model.
- *shape*(x,y,z), which provides the individual shape of a particular component. In Figure 2, I used two sphere-shaped components.
- *space*(x,y,z), which provides for spatial effects—rotation, scaling, stretching, and so on—for each component.

The MGSM model is implemented in Silly Space by taking each of these quantities and tying them together in a fast C++ implementation (see Listing One; listing begins on page 81). The exact model for the MGSM is given in Figures 4(a), 4(b), and 4(c). The first step is to perform the preliminary rotation of an individual component in the field. For instance, if you have a normal sphere component and a sphere component that has been stretched in two directions into a disc-shaped object, you could rotate the disc freely in any number of directions. This rotation is specific to the component only, and all other components will stay fixed in space.

Figure 4(a) shows the equation form for the rotation that is the first step in the MGSM. Figure 4(a) is part of the *space*(x,y,z) transformation. The rotation equation takes the three angles, relative to the x -, y -, and z -axes, as well as the parameters given as arguments to the MGSM call, *i*(x,y,z).

Now that we have performed the rotation of the given testing point with the equation in Figure 4(a), the next step is to

Kyle is a member of the Advanced Technologies Group at Advanced Visual Systems in Boston. He is also the author of POWER-3D: High-Speed 3D Graphics in Windows 95/NT (Manning/Prentice-Hall, 1997, ISBN 0-13-841214-6). He can be contacted at lussier@inside.net.

add stretching and translation to the individual component. This transformation is also still part of the $space(x,y,z)$ aspect of the MGSM.

The last modeling aspect of the MGSM is to apply the shape of the current component. Figure 4(c) is one example component. The shape given is the implicit equation of a sphere (metal-ball). The evaluation of this quantity is for a single component in the entire surface. The corresponding summation of many of these components is what makes a complete, artistic surface in Silly Space.

Decomposing MGSM into Triangles

Silly Space makes use of Jules Bloomenthal's implicit surface tessellator, which was presented in *Graphics Gems IV*, edited by Paul Heckbert (Academic Press, 1994). The tessellator works by accepting a function similar to the MGSM and searching the space to find the edge of the surface.

Bloomenthal's tessellator moves around on the surface of a geometric definition and creates a triangle-based tessellation of the model. It does this by using a collection of cubes to walk around on the surface. Each cube is used to isolate the surface along the boundaries of the cube (see Figure 5). The resultant testing of the 8 vertex locations on the cube gives $2^8=256$ possible test cases. These test cases are either positive (inside the surface) or negative (outside the surface). The test cases are mapped to bits and used in a look-up table to determine which edges need to be searched for the surface. After calculating all of these edge/surface intersection points, the algorithm then spits out the representative triangle set for that cube. The cube is then moved to the next point on the surface.

Silly Space uses a hybrid version of Bloomenthal's original tessellator. Some specific additions and modifications to it for real-time interactive use include:

Initial surface detection. The original tessellator used a random space searching method to find the starting edge of a surface. Silly Space computes the exact starting point of a surface and begins tessellation at this point. This turns out to be much faster than the random searching method if you already know where to start.

$$f(x,y,z) = 1.6 - (0.8) \frac{1}{(x+15)^2 + y^2 + z^2} - (0.8) \frac{1}{x^2 + y^2 + z^2}$$

Figure 1: Function that describes the implicit surface of a two-component blob.

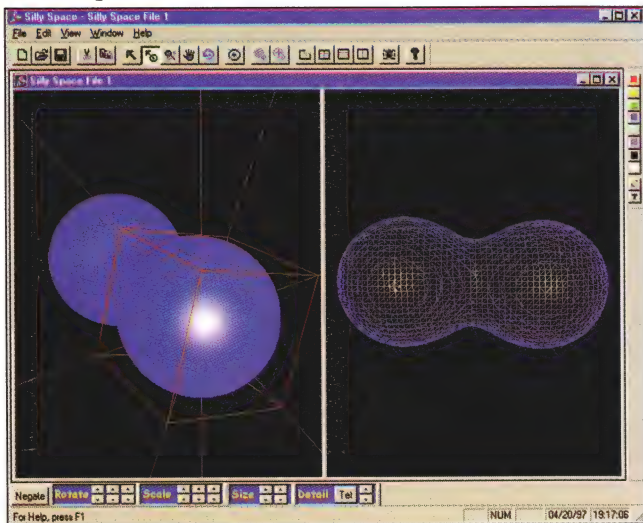


Figure 2: Two-component blob modeled in Silly Space.

Object orientation. The tessellator was broken down into a modular object-oriented form in C++. This new modular form was ideal for finding optimization points, and code enhancements gave an overall performance increase.

Disjoint surface detection. The original tessellator did not find disjoint surfaces, or surfaces that were not attached to the currently tessellated surface. Silly Space's implementation detects and tessellates these disjoint surfaces in a nearly linear manner.

Improved memory model. The original memory allocation model was not well suited to repetitive real-time calls to the tessellator. The memory model was replaced with an allocation model that prevents memory leakage and speeds up overall memory allocation and destruction.

Component Shapes

Any given surface or set of disjoint surfaces is made up of a collection of components. These components can take many different shapes. In Figure 2, for instance, the two components were in the shape of spheres.

Other forms are possible, such as a cube, or even a toroidal-shaped component; see Figure 6. There are no real limits to the

$$i(x,y,z) = \text{Threshold} - \sum_N \text{shape}(\text{space}(x,y,z))$$

Figure 3: MGSM general form.

(a)

$$x' = \cos(\theta_y) \cdot \cos(\theta_z) \cdot X - \cos(\theta_y) \cdot \sin(\theta_z) \cdot Y + \sin(\theta_y) \cdot Z$$

$$y' = X \cdot \sin(\theta_x) \cdot \sin(\theta_y) \cdot \cos(\theta_z) + Y \cdot \cos(\theta_x) \cdot \sin(\theta_z) - Y \cdot \sin(\theta_x) \cdot \sin(\theta_y) \cdot \sin(\theta_z) + Y \cdot \cos(\theta_x) \cdot \cos(\theta_z) - \sin(\theta_x) \cdot \cos(\theta_y) \cdot Z$$

$$z' = -X \cdot \cos(\theta_x) \cdot \sin(\theta_y) \cdot \cos(\theta_z) + X \cdot \sin(\theta_x) \cdot \sin(\theta_z) + Y \cdot \cos(\theta_x) \cdot \sin(\theta_y) \cdot \sin(\theta_z) + Y \cdot \sin(\theta_x) \cdot \cos(\theta_z) + \cos(\theta_x) \cdot \cos(\theta_y) \cdot Z$$

(b)

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} (x' + T_x)S_x \\ (y' + T_y)S_y \\ (z' + T_z)S_z \end{bmatrix}$$

(c)

$$\text{Contribution} = \frac{R}{(x'')^2 + (y'')^2 + (z'')^2}$$

Figure 4: (a) Rotation component of $space(x,y,z)$ in MGSM; (b) Translation and Stretching component of $space(x,y,z)$ in MGSM; (c) implicit sphere equation in shape component of MGSM.

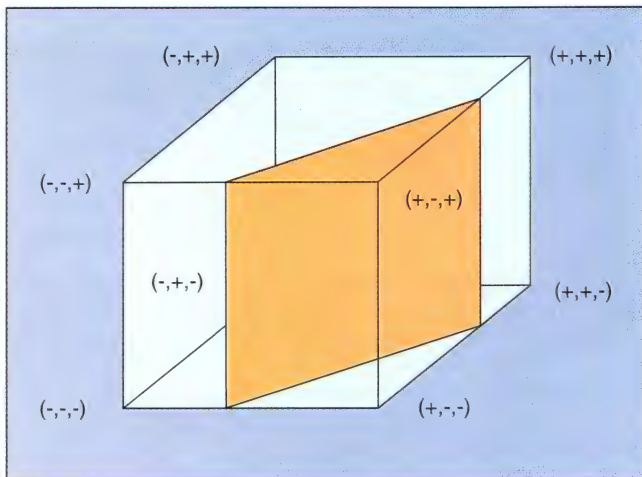


Figure 5: Isolating the surface along the boundaries of the cube.

variety of individual field-based components in Silly Space. The trick is finding the ones that are computationally quick to evaluate and are actually useful as primitive modeling components.

Spatial Transformations on Components

Another significant part of the MGS is the idea of applying various spatial effects to any given component shape. Spatial-transformation effects are applied to part of the model. All traditional spatial transformations fit well into this model.

For instance, scaling along the x-,y-,z-axes can be done by modifying the spatial-transformation part of the model so that it scales the individual component along each axis. This could be done, for instance, by multiplying the testing point by, say, 0.6 to shrink the model on the x-axis by 40 percent. Figure 7 shows five individual components—one a perfect sphere, the others stretched into the forms of disks. Each disk is created by taking the perfect sphere and stretching it along two separate axes. Additionally, each disk is rotated about different axes. The wireframe camera view on the right has lines indicating the centers of each component.

Components can also be created that subtract space from a surface. For instance, consider using a number of components in the traditional sense (each component adds to the model). If you take a single component that is among a collection of positive or additive space and invert the field, the component will subtract from the space of the model.

On the front cover of this issue, for instance, the penguin's nostrils use negative or inverse space to subtract from the main figure. Mixing and matching both positive and negative space provides a powerful way of shaping and molding the space. This combined with different component shapes and the transformations you can apply to these shapes gives the user a flexible, easy to use modeling application.

Texturing Implicit Surfaces in Real Time

Silly Space also provides a real-time texturing capability for implicit surfaces. In Figure 8, the human hand was created in about 15 minutes by stretching and shaping each finger and attaching them to a base disc. A texture that resembles skin is then attached to the surface. Note the top-right camera view. This view is inside of the hand looking down the finger spaces. The lines are the component fields that make up the hand. The texturing is done by importing and attaching a Windows bitmap (BMP), a device-independent bitmap (DIB), or a run-length encoded bitmap (RLE). Because the image file is converted to one of these formats, any image can be used.

Silly Space implements a couple of optimizations to speed texturing of the surfaces. The first step is to reduce the given texture to a power of two and scale it into a square. Computers can multiply and divide by a power of two very quickly. The multiplication or division by two operations boil down to a shifting of bits, as opposed to a multiplication call. For instance, the number three is 11 in binary, and the number six is 110. By shifting the bits left one, the number is doubled. This aids the scaling process as the bitmap size can be scaled to a good texturing size very quickly. The closer the size of the available texture to the one being drawn, the better the texture mapping. Silly Space also converts the bitmaps palette to a true color 24-bit palette. Figure 9 shows the complete hand with the texture color blended with the base color of the implicit surface, which is in this case a light orange.

The Silly Space API

Silly Space uses the traditional MFC document/view model; see Figure 10. Silly Space provides derived classes from the base *CView* and *CDocument* classes. These derived classes are called *CSillyView* and *CSillyDocument*, respectively.

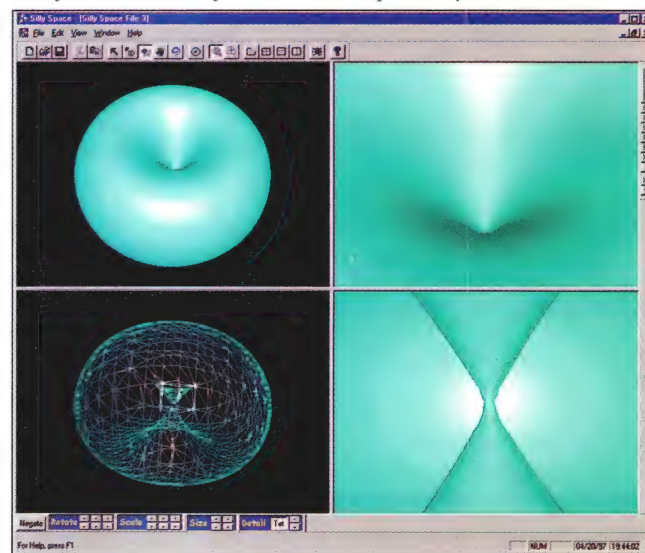


Figure 6: Toroidal-shaped components.

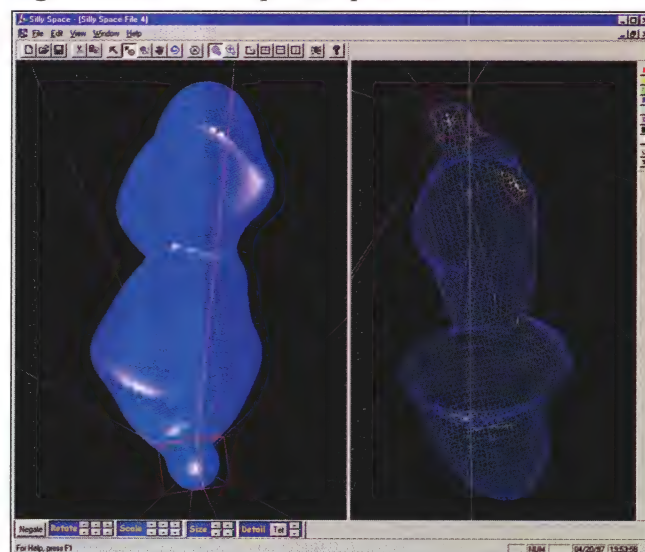
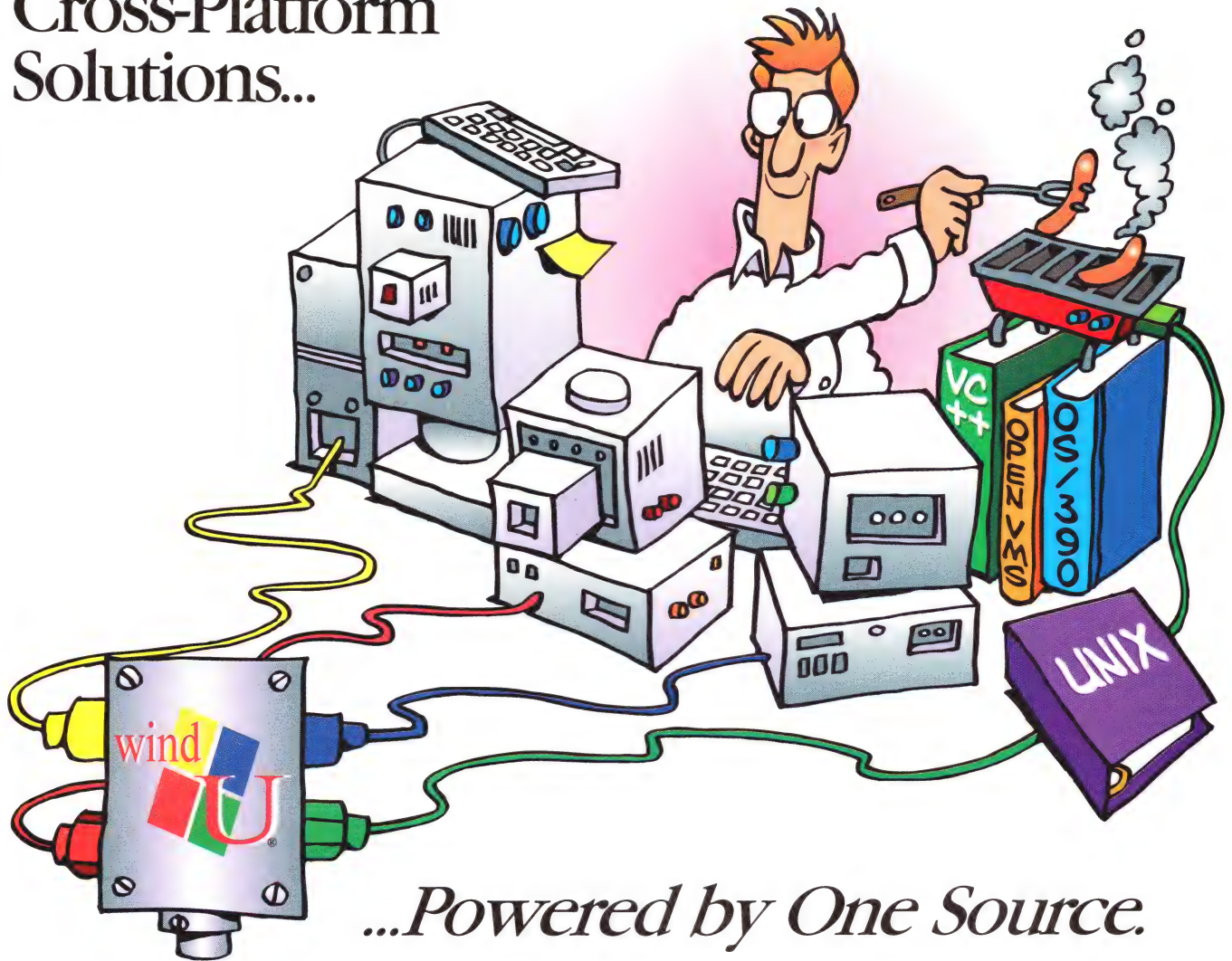


Figure 7: Five components—a perfect sphere and four disks.

Cross-Platform Solutions...



...Powered by One Source.

Get a power-surge from your Windows code! With Wind/U® from Bristol Technology®, you can re-use your Microsoft Windows source code to deliver applications across multiple platforms in the enterprise.

Interested in increasing productivity? Spend more time creating or enhancing your core software products. Wind/U lets you produce a functionally identical UNIX, OpenVMS, or OS/390 native version in a fraction of the time and cost.

Do you need ActiveX Controls on UNIX? The new Wind/U ActiveX SDK allows you to re-use your ActiveX source code with Internet Explorer for UNIX.

To ensure compatibility with the most recent Windows releases, Bristol licenses Microsoft Windows source code and incorporates this technology into Wind/U.

Leverage the power of Wind/U, the Windows API, and MFC today. You'll get increased productivity. You'll get superior functionality. And you'll get a cross-platform application powered from a single source.

Wind/U offers:

- Win32 API & MFC on UNIX, OpenVMS, and OS/390
- The latest Visual C++ features
- Complete OLE and ActiveX
- Threads and Winnet APIs
- Native look-and-feel & performance
- Windows Common Controls & Dialogs
- Industry standard online help (WinHelp) & PostScript/PCL printing

Get more information on Wind/U:

www.bristol.com or info@bristol.com

In the U.S. call:
(203) 438-6969

In Europe call:
+31 (0)33 450 50 50



BRISTOL
T.E.C.H.N.O.L.O.G.Y

Delivering Cross-Platform Solutions.

Wind/U and Bristol Technology are registered trademarks of Bristol Technology Inc. All other products mentioned herein are trademarks of their respective holders.

His Crime ?



He spent months writing Fax support for his Windows application.



Using **FAXMAN** he could have done it in minutes.

FaxMan provides Windows Developers with a Plug-In programmable fax engine which instantly allows applications to send and receive faxes using standard off the shelf fax modems.

The FaxMan SDK includes 16 & 32 bit DLLs and ActiveX controls as well as Windows 95 and NT printer drivers for creating fax output. And because FaxMan was designed for programmers, not end-users, you provide the user interface so your users won't even know you're using it.

Visit our website and download the complete FaxMan documentation and you'll see how FaxMan can simplify your development.

**To purchase FaxMan online,
visit our website at
www.data-tech.com
or call
800-955-8015**



300 Pensacola Road • Burnsville, NC 28714
704-682-4111 • Fax 704-682-0025 • BBS 704-682-4356
Email: sales@data-tech.com

FaxMan is a trademark of Data Techniques, Inc. Windows is a registered Trademark of Microsoft, Inc.

CIRCLE NO. 80 ON READER SERVICE CARD

(continued from page 20)

The Silly Space document object, *CSillyDocument*, contains all of the data necessary to build the live model. This includes the surfaces, spaces, shapes, and tessellations of the model. It also includes the necessary streaming code for writing the models and surface bitmaps to disk. The object handles all of the following details:

- Streaming of objects to a *CArchive* object for persistent (on disk) storage.
- Collections of components with the associated parameters for computing the geometry surface of a model.
- The shapes (sphere, disk, and so on) of user-created components in the model.
- The spatial transformations (scaling, translation, and the like) of user-created components.

Silly Space makes use of Jules Bloomenthal's implicit surface tessellator

The Silly Space view object, *CSillyView*, provides the implementation details of the 3-D user interface. It also handles construction and setup of the OpenGL rendering context, or the HRC. Command handlers provide control and manipulation of the modeling surfaces. For instance, you click on one button to increase the detail level of the surface tessellation. This changes the necessary geometry and *CSillyDocument* parameters and recomputes the tessellation. *CSillyView* is responsible for:

- 3-D click-and-drag UI, which is what makes Silly Space so easy. It makes use of Geometry Object Extensions (GOEs) to handle the intersection and interactive manipulation of components in the space.
- Command handlers, which provide input and capability beyond the 3-D click-and-drag UI. Command handlers for the view change the current tool (from a node select tool to a hand tool, for instance).
- Camera view parameters. Individual views have different camera viewing parameters. Each view can be a different camera angle on the model.
- OpenGL initialization. *CSillyView* also provides complete initialization and setup of OpenGL for proper use in Silly Space.

Geometry Object Extensions

OpenGL provides a framework for fast and accurate graphics, but it does not do everything that is necessary to make Silly Space work. The biggest deficit is the lack of an internal method for holding and maintaining data containers and data structures for building models. This is not really a downside of OpenGL, however, as every graphics application will need a different form of container. Silly Space requires the

Dr. Dobb's Journal, July 1997

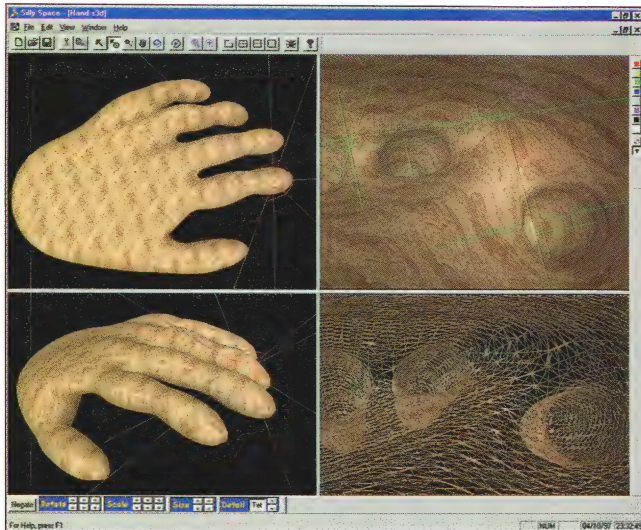


Figure 8: A hand with a texture attached to the surface.

ability to collect data and then provide user interface queries, creation and deletion of members, fast pipelining of data into OpenGL, and so on. To provide these abilities, I created Geometry Object Extensions (GOEs) to the OpenGL framework. GOE objects are what make Silly Space easy to use. They provide an abstract layer of communication between the surface tessellator, the object database (*CSillyDocument*), OpenGL, and the user. The objects I created include:

- *LObj*, a derived object from the MFC *CObject*. This object specifies proper virtual methods for drawing an object and for persistently storing the object.
- *LVector*, the fundamental 3-D math geometry object. This object provides normalization, magnitude, scaling, skewing, and other members for handling 3-D space.
- *LPlane*, the plane object that handles all of the intersection and collision functionality. This object is the fundamental unit for the 3-D UI.
- *LCube*, a cube object that can be drawn, and is used in in-



Figure 9: Texture color blended with the base color of the implicit surface.

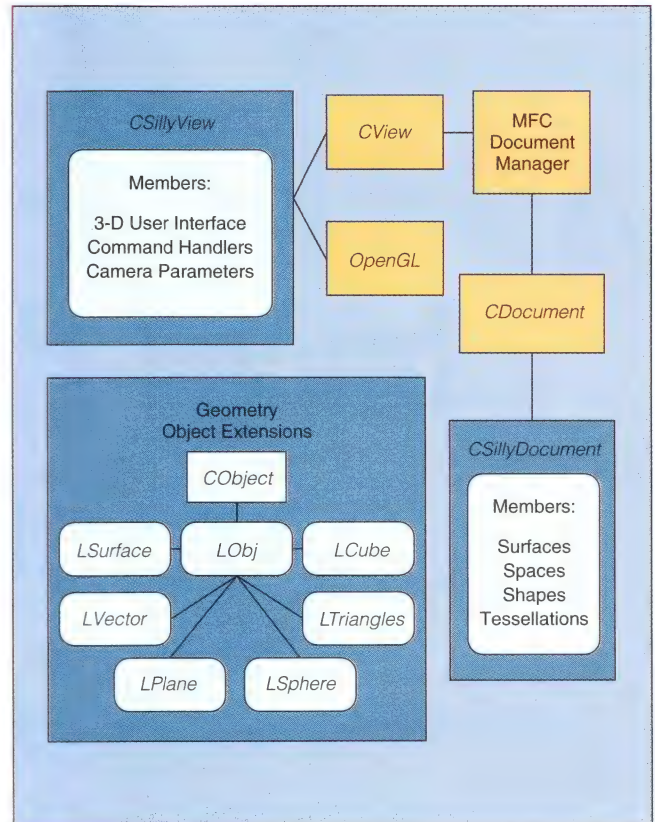


Figure 10: Silly Space's implementation of the MFC document/view model.

tersection calculations with the mouse. The cube object uses a set of six *LPlane* objects for intersection testing.

- *LSphere*, which can be drawn like the others and provides a ray-to-sphere intersection capability. Stretched spheres are used for intersection of the implicit surfaces in Silly Space.
- *LTriangles*, which provides organization of vertex-indexed triangles and handles the drawing and modeling of collections of triangles in an API optimized manner. For instance, this object uses the shared vertex lists feature (when possible) that is new in OpenGL 1.1.
- *LSurface*, which encapsulates the implicit surface tessellator, intersection calculations for the user interface, different drawing modes, texturing, and the overall behavior of surfaces in Silly Space.

Spline Paths for Components

Suppose you want to move the components of the field around in space. You know those movies they make with lumps of clay? They move the clay and then take a picture, and then move the clay again, and so on. Eventually they have a complete movie. This process is slow and cumbersome.

Silly Space lets you design the complete model, and then key-frame motion in the model. You can, for instance, create three separate states for the model. Silly Space then takes the user parameters of the quantity and fits spline curves through these parameters. The result is an infinitely smooth animation of the components, even though the user only specified three frames.

DDJ
(Listing begins on page 81.)

A Memory-Constrained Image-Processing Architecture

Patterns for processing pictures

Mayur Patel

Image-processing applications are hungry for both processing time and memory. CPU demand typically does not pose a problem, since most operating systems' schedulers can adequately balance the needs of competing applications. However, virtual memory is the only service provided by the operating system to deal with the memory load. Allowing an image-processing application to swap memory pages to disk impairs all the activity on the machine.

A good image-processing library should deal with this memory problem with minimal supervision from you. In this article, I present an image-processing architecture that gives you a reasonable degree of control over memory consumption.

Tile On Demand

Tile-based processing consists of dividing an image into small, rectangular pieces called "tiles," processing each tile, and re-

Mayur is a member of the software development staff at Cinesite Digital Film Center, a full-service post-production and visual effects facility. He can be contacted at mpatel@cinesite.com.

assembling the image. The principle of locality applies, so an image-processing operation usually requires input tiles of approximately the same area as the output tile. Clearly, processing each tile requires less memory than processing the entire image.

Many existing image-processing libraries support scan-line-based rather than tile-based image processing. Scan-line libraries



are easy to implement and often perform efficiently because most image file formats store data in scan-line chunks. However, a tile-based approach retains the same advantages (scan-lines are simply tiles with a height of one pixel) and is more general.

One case where you benefit directly from flexible tile size is with transform-based image compression. Many popular Discrete Cosine Transform-based compression algorithms require 8x8 tiles. With

scan-line-based libraries, you must construct 8x8 units by writing eight scan lines into a buffer and moving data back and forth from the buffer accordingly. This is unnecessary busy work; it is better to provide tiles in whatever proportions you request.

Another convenient feature is lazy evaluation. Processing on demand reduces the expense of image processing and promotes interactivity. For example, users may reorder the sequence of operations several times before explicitly requesting an output image. Users may also require only a small portion of the output image or may extract data somewhere in the middle of the pipeline. Such requests can be serviced in less time than it takes to produce the entire output. Demand-driven processing helps to minimize wasted computation in these situations.

In this framework, data requests propagate from the output toward the input (see Figure 1). The input satisfies those requests and triggers the appropriate computation, and the result flows back toward the output. Along the way, operations are applied to the data according to the dependency information already supplied through the application. By the time the data reaches the output, all the specified operations have been performed.

Coupling demand-driven execution with tile-based processing yields an environment with great potential. Certainly, these features carry with them nontrivial overhead. However, they also work together to reduce the total amount of memory required to process images. The extra cooking time is often worth the wait.

PowerBar energy bar flavors- chocolate, malt-nut, wild berry, apple-cinnamon, banana, mocha, peanut butter, oatmeal raisin ©1997 Powerfood, Inc. 800-55-POWER, In Canada- 800-500-5634, www.powerbar.com



Brian Maxwell, Founder & CEO - Powerfood, Inc.

-founded PowerBar®
in Berkeley, California
-on his way to meeting
in Chicago

BALANCED NUTRITION
LASTING ENERGY™

note dee-licious
new flavor



PowerOn™

Better Productivity Tools!

TE Developer's Kit

Rich Text Edit Control

This advanced Rich Text

control features **wysiwyg** multiple fonts and colors; paragraph indentation, justification, spacing, tabs, border and shading; RTF import/export, embed pictures, controls, and OLE objects, read-only mode and printing.

Advanced features: Pagination, columns, sections, hyperlink support, hidden and protected text, tables, page header/footer, text/picture frames, line/boxes, print preview with zoom. Includes DLL, VBX, OCX, and MFC interface. Also includes the complete 'C' source code. (Windows or WIN32: \$389).

Call for DOS & OS2 versions and an **HTML** viewer/editor control add-on (\$299)

ReportEase Plus

ReportEase Plus consists of a report layout editor and a report executer. Advanced features include: multiple files, multiple sorts; data, calculation, system, and dialog fields; bold, underline, italic formats; subtotals, record filter, functions, word wrapping and more. This DLL/VBX features a **graphic form editor with line/box drawing, colors/shades, drag/drop, print preview, etc.** Includes the 'C' source code. (Windows or WIN32: \$459)

Form Plus

Form Plus consists of a form designer and a form executer. The product supports **line/box drawing, fonts, colors/shades, drag/drop, check box, radio button, push button, images, print preview, etc.** (Windows or WIN32: \$399).

Spell Time

Spell Time consists of a dictionary (over 100K words) and the routines to access the dictionary. It also includes an application specific dictionary, and a user dictionary. The routine will suggest alternatives for a misspelled word. Highly optimized: 400 to 500 words/sec on a 33 Mhz 386 computer. Includes the 'C' source code. Specify *DOS, Windows, WIN32 or OS2-PM.* (\$389)

Other Products

ChartPro: This DLL draws bar, pie, line, area, xyz point chart (\$399). **Rich Text Grid:** This grid control supports multifont and multiline text within each grid cell (\$399).

Sub Systems, Inc. 800-447-6819

Fax: 508-352-9019

Demos available: www.subsystems.com

11 Tiger Row, Georgetown, MA 01833, 508-352-9020

CIRCLE NO. 157 ON READER SERVICE CARD

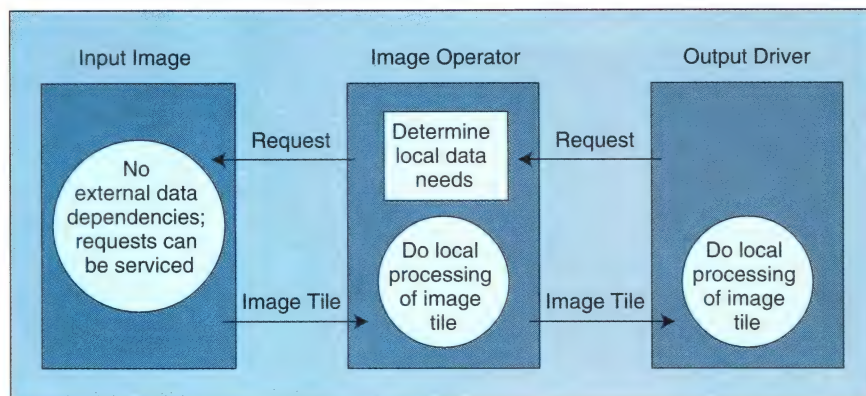


Figure 1: Tile-based, demand-driven processing.

(continued from page 24)

Object Interaction

This tile-based image-processing architecture is inherently object oriented. The interaction between three fundamental classes—*Image*, *ImageTile*, and *ImageMan*—is the foundation of the system.

Tiles constitute the interface between the library and the application. You need not be aware of the internal representation of the image as long as the tile implementation is sufficiently general to handle the widest range of cases.

Applications request pixels in two ways. The more convenient interface is *Image::newTile()*, which allocates a new tile and fills it with image data. The other interface is *Image::fillTile()*, which accepts an existing tile for processing and overwrites its contents. The API is as simple as that.

Behind the scenes, the image interacts with the image manager; see Figure 2. *Image::newTile()* requests (which go to the image manager) let the image manager provide a host of services, including caching.

The manager eventually comes back to the image and uses its *Image::fillTile()* method to compute data.

In the obvious implementation of this interaction pattern, there is one inefficiency. When the system receives a *newTile()* request, it immediately allocates memory. *Image::fillTile()* then pulls input data into scope for local processing. The *fillTile()* method most likely uses the *newTile()* interface to request data, so memory is allocated at every step as requests propagate backward. These allocations aren't actually needed until the data begins to flow forward.

A more constrained approach allocates memory only after the input data is available, but I prefer the former behavior for two reasons:

- It supports a flexible interface between the image and image manager.
- It supports a simple developer interface for the implementation of real image processing operators.

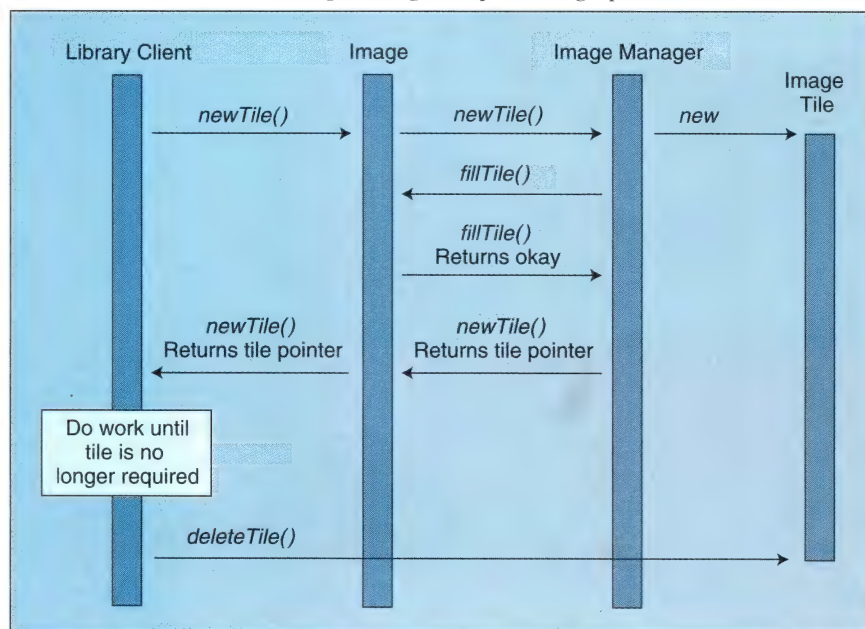


Figure 2: Object interaction.

Implementation

The tiles in my architecture use the (X,Y,Z,C) image-space model. The X component represents the width, Y represents the height, C represents the color channel, and Z the frame number. This model lets you describe images of any number of channels. It also lets you describe a series of images as a sequence, which is useful for processing digital video or film since an animation is expected to consist of several frames, each with a uniform width, height, and channel count.

The coordinate (x,y,z,c) is sufficient to specify the smallest datum. In addition to these coordinates, my tile structure (see Figure 3) contains width and height elements that make it possible to represent a one-channel area from an image. You can access pixel data using methods provided by the tile class.

Different applications will require pixels of different precision. For example, 8 bits per color channel commonly represents image data. For higher quality work, however, 16 bits per channel is more useful. In some cases, particularly in scientific computing, 32-bit floating-point values (0.0 to 1.0) best represent each plane. This implementation supports all three options. A method allows typecasting the data between supported types.

Granted, using macros in C++ is generally bad programming style. Likewise, cute macro tricks in C also represent poor programming style. However, I have used both to encapsulate data-type support. Any technique, with disciplined execution, has its place. In this situation, a new data type may be incorporated with the addition of only one function. No changes to existing code are necessary. You can judge the case for yourself. This approach is discussed in more detail in the accompanying text box entitled, "A Macro Technique for Managing Types."

The tile class inherits the properties of a referenced object. Smart pointers may be used to organize tiles at the application level. The method `Tile::deleteTile()` is included to complement the `Image::newTile()` method. `Image::newTile()` provides construction services; `Tile::deleteTile()` provides dereferencing and destruction services.

My image manager implementation is generic, providing the minimum set of services required to meet its obligations. At most, only one image manager may exist at any time. An auxiliary class regulates access. The Singleton design pattern applies well to the image manager, making sure only one instance of a class exists at any time. Blending the referenced-object pattern with the Singleton pattern

also ensures that the image manager is deallocated when it is not required. This is useful if image processing represents only a small portion of the total application. Attaching an image-manager reference to images through the base class ensures that the image manager is instantiated during the lifetime of any of its potential clients.

I provide two base classes for images: *InputImage* and *SampledImage*. *InputImage* best suits images (described as mathematical functions) that may have variable resolution or unlimited range. *SampledImage* best suits images of finite resolution and range.

SampledImage uses the Strategy design pattern to deal with out-of-range behavior.

The Strategy design pattern encapsulates implementations of an algorithm so that the algorithm isn't hardwired into a class. A *SampledImage* may receive a request during a convolution calculation for data beyond its range of valid samples. Fill-strategy objects may be attached to *SampledImage* to determine how this area is validated before the tile is returned to the client.

I don't use a base class for the output image. Since the output can flow to the screen or printer as easily as to a file, the output mechanism will often depend upon application-specific code. Additionally, the demand-driven characteristic of the architecture makes it easy not to place constraints on the output.

The longest
continuously
advertised C product
in the history of... mankind.

PC-lint

for C/C++
Version 7.0
presents Bug # 1404

```
1  #include <iostream.h>
2
3  class X;
4
5  void eradicate ( X * p )
6      { delete p; }
7
8  class X { public: virtual ~X(); };
9  class Y : public X { public: int ay; ~Y(); };
10
11  X::~X() { cout << "X deleted\n"; }
12  Y::~Y() { cout << "Y deleted\n"; }
13
14  int main()
15  {
16      Y *p = new Y;
17      eradicate( p );
18      return 0;
19  }
```

The programmer expected some deletion notification messages to appear (at runtime) but instead got nothing. How come? Call if you need a hint or refer to our web page at www.gimpel.com

PC-lint for C/C++ will catch this and many other bugs. It will analyze a mixed suite of C and C++ modules to uncover bugs, glitches, quirks and inconsistencies.

Version 7 of PC-lint breaks new ground with inter-statement value tracking for both automatic variables and class data members. Taking clues from assignment statements, initializers and conditional expressions it can detect out-of-bound subscripts and potential null pointer uses. As an enabling technology, almost 100 standard functions are rigorously checked. Also macros are subject to increased scrutiny, checking for unparenthesized parameters, unparenthesized bodies and repeated arguments having side-effects.

Plus Our Traditional C/C++ Warnings:

Uninitialized variables, inherited non-virtual destructors, strong type mismatches,

inadvertent name-hiding, suspicious expressions, etc., etc.

Full C++ Support - PC-lint for C/C++ is based on the ARM and is tracking the latest ANSI/ISO draft including exceptions and templates. It supports both Borland and Microsoft C/C++.

PC-lint for C/C++ \$239

Numerous compilers/ libraries supported. Runs on MS-DOS (Optional built-in 386 DOS extender), OS/2, NT and Windows 95. This price is subject to increase.

FlexeLint for C/C++

The same great product for other operating systems. Runs on all Unix systems, VMS, mainframes, etc. Distributed in shrouded C source form. Call for pricing.

Gimpel Software

3207 Hogarth Lane, Collegeville, PA 19426

CALL TODAY (610) 584-4261 Or FAX (610) 584-4266

30 Day Money-back Guarantee.

PA add 6% sales tax.

PC-lint and FlexeLint are trademarks of Gimpel Software

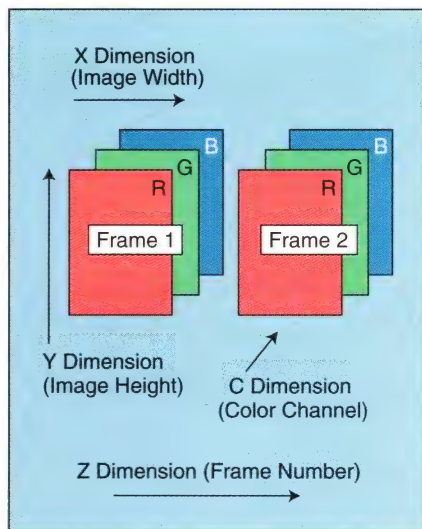


Figure 3: The (X,Y,C,Z) image-space model.

Example Applications

Listing One (listings begin on page 81) is a test program that enhances the contrast of an SGI RGB file. The program first specifies data dependencies using class constructors. The first object opens an SGI RGB-format file. The second object executes the contrast-enhancement operation. The last object writes the data to an output file, possibly converting the data type. Once these dependencies are defined, instantiating the output image triggers processing of this simple chain.

SGI RGB files are both powerful and flexible, supporting 8- or 16-bit integer

channels and any number of color channels. As you can see from the *Contraster* class (see Listings Two and Three), implementing new operators is straightforward. *Contraster* consists of a constructor, a destructor, and a *fillTile()* implementation. Developers of new image-processing operators are well shielded from the image manager and other system complexities.

The *Contraster* object executes the contrast-enhancement operation using floating-point math. I did this for two reasons. First, it demonstrates how easy it is to typecast the data in tiles. (But be warned: Typecasting is as expensive as any pixel-processing operation.) Second, using floating-point math makes it easy to deal with over- and under-exposure problems. Integer values may overflow or underflow, whereas floating-point values retain data integrity. There are ways around integer saturation, but I show floating-point in action for the sake of example.

Extending the Library

Since my image-manager implementation doesn't take full advantage of the flexibility of the architecture, it is a good place to start extending my implementation. One of the first features that should be incorporated into the image manager is a tile-caching mechanism. Imagine a chain of convolution operations executed using the current implementation. As the requests for tiles propagate toward the input, the size of the requested area

becomes larger. Over time, these requests will result in excessively redundant calculations.

A cache minimizes redundancy by storing data that is perceived to be frequently used. Although the cache requires more RAM—one of the constraints you're trying to overcome—you can control how much RAM it uses. Giving some RAM back to the processing engine for caching is a guard against extremes in the space-time tradeoff. Remember, caching operator data saves more than the computation at that node in the pipeline. It saves computations at every node up to and including that operation.

How you implement the caching mechanism depends on the execution characteristics that you want the library to have. You can cache individual tiles, color channels, or entire images. You may need predictive caching or reactive caching. Matching your needs with a cache implementation is well worth the development time.

Earlier, I described why I have two image base classes. The *InputImage* encapsulates infinite-range or infinite-resolution data. The *SampledImage* encapsulates traditional image representations. If the caching technique you prefer depends on the image's resolution or range, you will not be able to cache *InputImage* images. If so, then you may add *InputImage* support by making *InputImage*'s *newTile()* method defer to the image manager. This gives you more flexibility in experimenting with new caches.

A Macro Technique for Managing Types

In the image-processing architecture presented here, I use a macro technique to encapsulate data type support. Like any macro technique, it requires disciplined use. However, it provides a convenient way to make branching decisions based on type information.

C/C++ programmers are familiar with the source-header file combination. To implement this macro technique, I need another kind of file. I call this a macro file, and I use a ".m" extension. Some choose to call it an include file, with a ".i" extension.

There are three things to notice in Listing Four, which is an example macro file I call "Type.m." First, all macros are undefined by the end of the file. Second, the macro *TYPEMACRO* is not defined within this file, even though it is used. *TYPESEPER* and *TYPEENDOFLIST*

are defined only if they haven't already been instantiated. Third, one of the elements in the *TYPEMACRO* is a formal data type.

The parameters to the *TYPEMACRO* calls encapsulate a table of information. Depending on your application, you will need different information in this table. The trick is to keep this table minimal. In the image-processing architecture, I had to include function names, numeric values, and precedence information.

Listing Five shows one way you can use the macro file in Listing Four. This bit of code declares an enumeration that can be used to identify and manage types in a cleaner way in other parts of the code. The *TYPEMACRO* definition allows information to be extracted from the table in the *Type.m* file and used in the current context. The *TYPESEPER* and *TYPE-*

ENDOFLIST elements allow the included table entries to adapt to the syntax of their context. Listing Six illustrates another way to use the macro file in Listing Four.

The application uses the enumerated type declared earlier to provide type information when the function is invoked. In this case, data is available through void pointers, but in general, any interface that supports easy typecasting will work. The macro definition and *include* statement fill the switch with a variety of alternative behaviors, depending upon the specified type.

I originally learned this technique while implementing databases, but it clearly can be applied to a wide variety of situations. With sparing, disciplined use, it can be a powerful tool for managing types in C/C++.

—M.P.

Once you have implemented a cache, it would be nice to offer forced caching. Some image operators, such as a Fast Fourier Transform (FFT), are not localized. To execute an FFT, it is easiest to process the entire image at once. And it isn't cheap. Forced caching lets you identify and address hot spots for the caching system, such as these nonlocalized operators. Without forced caching, or if there isn't enough room to cache the entire FFT image, each tile request arriving at the FFT image will likely result in recalculation of the entire image.

Another characteristic of the current image manager is that tile sizes are not regulated. Currently, the application is on its honor not to request more data than it actually needs at one time. Poorly designed applications may request the entire image in one call. The current image manager will propagate this request. This defeats the purpose of having an architecture designed to constrain memory demand.

Request decomposition allows the image manager to decompose requests for large tiles into several requests for smaller tiles. Processing each smaller tile requires less memory than the whole. Another benefit of request decomposition is that the requests for the many small tiles may be executed in parallel. Giving the image manager the power to dispatch concurrent requests empowers the entire system with concurrent execution behavior. The developer of image-processing operators doesn't need to worry about writing parallel code.

Conclusion

The source code accompanying this article (see "Availability," page 3) implements the framework of this architecture. It is written in C++ and should be fairly easy to port. I encourage you to develop the architecture into an image-processing library custom fitted to your specific environment. For small systems, it should allow processing pipelines of considerable complexity. For multiprocessor systems, it should allow multiple renders to coexist without malicious contention for resources.

The Silicon Graphics Imagevision Library provides a superset of all the features I describe here, and I recommend that IRIX users evaluate it. However, it is not available for other platforms.

Acknowledgment

Thanks to Bob Amen and Cinesite Digital Film Center in Hollywood, California, for making equipment available for testing.

DDJ

(Listings begin on page 81.)

[A Continuing Discussion]

The Cool Objects Connection™



Business is Diverse

The Technical Resource Connection's objective is to architect information systems to reduce the complexity of **managing the diversity.** We do this through discussions. With business. Then...we do it. **For business.**



A WHOLLY OWNED SUBSIDIARY OF
PEROT SYSTEMS CORPORATION

800-872-2992 ■ Fax: 813-891-6138
<http://www.trcinc.com/> ■ e-mail: staffing@trcinc.com

Career Opportunities

Call, or fax or e-mail your resume to:
The Technical Resource Connection, Inc.
<http://www.trcinc.com/jobs/dd3>

CIRCLE NO. 337 ON READER SERVICE CARD

RaveKit: A Portable Graphics Framework

*A cross-platform,
renderer-independent
3-D toolkit*

Mark Carolan

Just as text-based terminals were an improvement over punch cards and Xerox-inspired GUIs were an easier interface for casual users, recent trends suggest a transition to 3-D depth-sorting to break down the complexity of multi-windowed applications. In this article, I present a general-purpose, cross-platform framework called “RaveKit” for creating interactive 3-D environments that provide an alternative way of presenting graphical data and obtaining user input. RaveKit demonstrates how you can use a multiplatform compiler (Metrowerks CodeWarrior Gold) to create a framework for producing high-resolution interactive 3-D graphics—without being tied to any particular commercially available renderer (QuickDraw 3D, DirectX, Renderware, and the like). I implement this renderer-independent strategy by deriving from a generic *Renderer* class and implementing the specifics in the subclass. The actual 3-D transformations and projections onto 2-D space are done on a vertex-by-vertex basis by the class *Vector3D*, while the prerasterizing processing of the output triangles is handled by a linked list of *Tri3D* ob-

jects that reference the vertexes of the world through indexes.

The RaveKit Framework

One of the purposes of the RaveKit framework (available electronically; see “Availability,” page 3) is to provide a simple API for 3-D programming without being tied to any particular renderer. To this end, I



provide a *Renderer*-derived class, called *Raver*, which uses Apple's RAVE API along with the default built-in RAVE engine. RAVE is the independent “immediate-mode” rasterizing module and Hardware Abstraction Layer (HAL) used by QuickDraw 3D.

The primary difference between “retained mode” and “immediate mode” is the amount of work left to the programmer. Retained mode provides storage of your 3-D world, shape generation, lighting, and so on, while immediate mode simply rasterizes the triangles you send it. This rule is not rigid, however. For example, Direct3D immediate mode offers lighting and a

few other services, while RAVE tends to stick to the bare essentials. While retained mode, with its larger feature list, is probably more appropriate to a 3-D scene builder or VRML designer/player, immediate mode offers the compactness and speed that is more appropriate to games or plug-in software.

The RAVE HAL is freely available from Apple (<http://www.quickdraw3d.apple.com/>) and it runs on PowerPC MacOS and 32-bit Windows. One of the most attractive features of RAVE is that it also serves as a standard 3-D API, regardless of the rasterizing software. This means third parties can provide an alternative RAVE engine while maintaining a consistent API. The RAVE documentation (available from Apple as a PDF document) provides specifications for creating your own engine, along with recommendations on which facilities must be provided and which are optional. For 3-D software and content developers, this removes the pain of having to learn a different API for each rendering engine. For engine developers, it offers a method of packaging a hardware-accelerated engine in a way that can be easily evaluated and incorporated into RAVE-savvy applications (including QuickDraw 3D).

Dropping in your favorite immediate-mode rendering library (as an alternative to RAVE) simply requires deriving a class from *Renderer* that implements the specifics of your renderer. For example, if Windows is your only required platform, you could call up the Direct3D immediate-mode API (which also offers rasterizing and HAL services, and is freely distributable). I have used triangles, as they are the most widely used primitive in commercial renderers. With some modification (mostly relating to clipping and depth sorting), you could incorporate quadrilaterals,

Mark is principal developer at HeadsOff Software and can be contacted at mark@headsoff.com.

Say Hello to the Next Generation C++ Toolkits

The ObjectSpace C++ Component Series leads the market in providing portable class libraries that solve today's biggest development problems. Designed by experienced Object Technologists, this series of 10 innovative C++ class libraries provides the foundation for all your C++ business solutions.

Version 2 of the C++ Component Series provides a full set of features ranging from the most portable implementation of the C++ Standard Library available today to libraries for systems programming and Web development. All libraries are portable across popular hardware platforms, operating systems, and compilers, and are safe for use in multithreaded environments.

By adopting ObjectSpace class libraries as the core of your C++ development, you can reuse large amounts of code saving both time and money. Because our class libraries are based on ANSI/ISO standards, you will not just reuse ObjectSpace code but standard interfaces as well. This will make your code vendor independent, increasing its reuse over time.

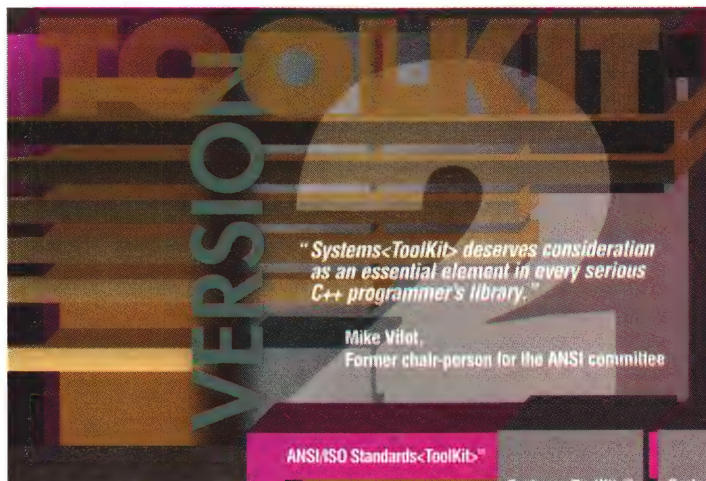
Systems<ToolKit>™

This power-pack contains eight of the ten ObjectSpace Component Libraries in one cost-effective package.

STL<ToolKit>™

This entry-level package contains four of the core ObjectSpace class libraries including the most portable version of the Standard Template Library available today.

Standards Compatible C++ Class Libraries for Portable Business Solutions



New Features

- Regular Expressions
- Portable Files and Pipes
- Hashing STL Containers

Improved Packaging

Packaged as ten independent libraries, bundled into three products.

Web<ToolKit>™

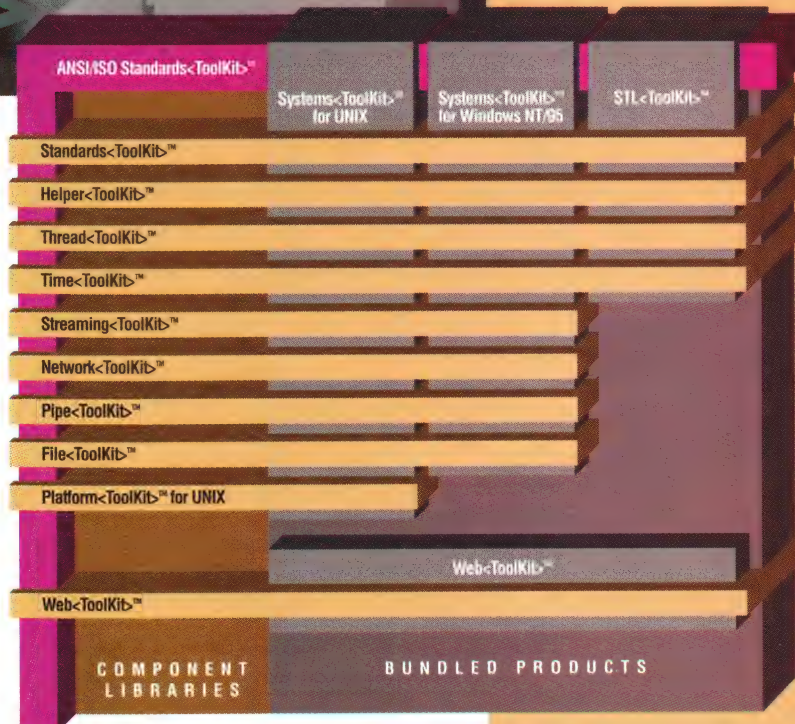
Add Web-based interfaces quickly and easily to your applications.



Products Include

All class libraries in the ObjectSpace C++ Component Series come complete with:

- Full source code.
- Extensive online or printed user guides and class reference manuals.
- Hundreds of examples.
- Full support for dynamic linked or shared libraries.
- No runtime royalties.
- **Free copy of STL for Java**



To order call 1.800.OBJECT.1, send email: info@objectspace.com,
or for more information visit our web site at
<http://www.objectspace.com>.

OBJECTSPACE

PRODUCTS • TRAINING • CONSULTING

14881 Quorum Drive, Suite 400, Dallas TX 75240, email: info@objectspace.com, fax: 214.663.9100, tel: 214.934.2496
ObjectStore™, OrbixWare™ and Visual Numerics™ are trademarks of their respective companies.

Wave Goodbye to
Proprietary Toolkits

More Platforms

Windows 95, Windows NT, OS/2
Warp, Macintosh, SPARC Solaris,
SunOS, Solaris x86, Linux, HP-UX,
IRIX, AIX, Digital UNIX, NCR UNIX

Better Integration

OBJECTSTORE

orbixWare
ORBITAL COMMUNICATIONS

Visual Numerics.

PREPARE FOR THE NEW STANDARD

CIRCLE NO. 172 ON READER SERVICE CARD

(continued from page 30)

ellipsoids, or other types as primitives. The heart of the rasterizing process occurs in the derived class's *TimeSlice()* function. *TimeSlice()* is passed an unsigned long containing bit values for any key combination returned from the *Scanner* class in *main()*. *Scanner* is a cross-platform asynchronous keyboard-scanning module, a frequently used tool in my class-library package. Note in Listing One (listings begin on page 82) that the RAVE calls are preceded by "QA," while RAVE data types are preceded by "TQA." Similar to the Direct3D COM interface, RAVE calls are implemented as macros that refer to the virtual functions within the various modules. The macros also act as a protection for the programmer against future modifications to the *vtable* structures.

Compared to RAVE, Direct3D immediate mode offers more services (for example, lighting and clipping). RAVE, on the other hand, offers the bare minimum, but is a much simpler API. Both make extensive use of floating-point arithmetic, and therefore require either the PowerPC (in the case of RAVE on MacOS) or Pentium on Win32 (not compulsory, but advisable). Both also respond well to hardware acceleration. A test I did on the program presented here on a prerelease Mac with the ATI RAGE chipset on the motherboard produced startling performance at 640×480×16, even in RaveKit's current unoptimized format.

The *Vertex3D* class (Listing Two) does a lot of the useful work and helps to delineate the abstract "world" rendering stage (done in *Renderer*) from the viewport projection stage. Having this clear demarcation between stages will make it much easier to incorporate extra functionality later.

The projection stage, handled by each of the vertex objects (which are actually aliases of *Vector3D* objects), is one of the most arbitrary in the process. Until everyone has PCs with hardware on par with a dedicated graphics workstation, real-time 3-D is the art of the possible. As programmers and designers, we are not necessarily concerned with physically accurate representations of reality, but are looking for a way to present our concepts as faithfully as resources allow. This usually means a constantly shifting compromise between ease-of-use, accuracy, detail, and frame rate. With RaveKit, I focus on ease-of-use and detail. The default pixel resolution is 640×480 with a 16-bit color depth. 16-bit color is, incidentally, RAVE's home turf.

The *Vertex3D* object also plays a part in the world transformation stage by transforming its own world coordinates according to instructions passed from the

Renderer class's *MoveWorld()* function; see Listing Three.

One obvious optimization to be done in the *Vector3DTransform* stage (Listing Four) is to create lookup tables for the sine and cosine values that are required if any rotations are necessary. This is straightforward, as there are currently only three possible increment values for interactive *Roll*, *Pitch*, and *Yaw*. For this article, however, I haven't bothered, because, with the relatively small number of vertices in this world, there is little visible difference. (On the Mac, I've used "LibMoto," Motorola's optimized PowerPC math library, which probably helps minimize the difference.)

For storing all the required vertices in the "solid" part of the world (that is, the vertices that move as one unit), I've used an array, allowing fast lookup from the linked list of *Tri3D* objects, which store the triangles as indices into the storage array (Listing Five). In a more demanding application, this array would need dynamic-sizing capability in, say, 1-KB chunks.

For demonstration, I've included a rudimentary mesh builder that allows an initial transformation and translation of a four-sided "panel" and a simple DXF importer. The DXF importer works on the POLYLINE/VERTEX combination (with the mesh specification) on DXF files exported from my little Swivel3D program, but you may need to tune it a little to read alternative DXF configurations. Since this program deals with triangles, it shouldn't be too difficult to get graphic data from other programs into this format.

RaveKit Enhancements

Adding a few bells and whistles to the RaveKit mini-renderer isn't difficult. I've simulated basic static lighting by calculating a light "strength" for each *Tri3D* object by reading a combination of distance-from-light-source and triangle surface normal, then tweaking the RGB values of each vertex accordingly. Colored lighting is basically there for free if you want it, by biasing light intensities toward the chosen color. Shadowing could be implemented with a bit of raycasting, but it just depends how much processing power you're prepared to donate. Texture mapping is already supported by RaveKit, and functions are provided in the *Raver* subclass for storing, animating, and attaching bitmaps.

While this approach doesn't offer the range of built-in features in a comprehensive renderer like Direct3D or QuickDraw 3D, there are some benefits to using an immediate-mode 3-D API, rather than relying on the more proprietary retained-mode interface:

- It keeps the top level API very lean, so incorporating new rasterizing technology doesn't require writing a lot of new calls.
- You can make more use of some of the excellent code examples provided by leading game developers and, of course, the reams of source code available on the Internet (it took me about 45 seconds to get information on the DXF file format for this article).
- A small package like this is very easy to install. The core program itself is tiny, and the RAVE run time is only around 120 KB. It can be dropped into the local directory, so you don't need to inflate the user's extensions or system folder with yet more DLLs that they will rarely use. This also makes it a candidate for a utility ActiveX component.
- Perhaps most important—especially for game developers—is the ease with which you can port to other platforms. As long as the API handles triangles, you can use the same code with a few customizations in your *Renderer*-derived class. This is of particular significance for porting to console platforms since, to my knowledge, there are no commercial retained-mode libraries that run on both consoles and PCs.
- Collision detection is practically automatic. Rather than duplicating a lot of the processing already done by a retained-mode renderer in order to sort nearby objects and surfaces, you already have this information, since each *Vertex3D* object has explicitly tested its relationship with the player.

Conclusion

Having explained the benefits of immediate mode over retained mode, there are also advantages to using an immediate-mode 3-D rasterizing engine instead of writing your own. These advantages include built-in hardware acceleration support, Gouraud shading, texture mapping, transparency, and z-buffering. (Actually, an optimization of this example program would be to switch off RAVE's built-in z-buffering, which may be overkill if your world is fairly simple, and provide your own instead.) And for those of you clever and resourceful enough to write your own rasterizing stage and HAL, there may be benefits to packaging it into a standard API specification such as RAVE and licensing it to graphics application developers.

The advantages of having that third dimension available, at relatively little extra cost in effort, are apparent once you start making use of it.

DDJ
(Listings begin on page 82.)

Now that APC Smart-UPS® includes FREE web-enabled PowerChute® *plus*, protecting network uptime has never been easier



November 19, 1996
APC Smart-UPS 1000

"...Inherent flexibility and excellent software... Don't be caught without one."



Power problems attack networks relentlessly. To protect hardware and data from system crashes, experts, network managers and computer users worldwide prefer one solution above all others combined: APC Smart-UPS. Now, all 120V Smart-UPS include FREE PowerChute *plus* power management software.

The most reliable protection you can buy

Smart-UPS provide complete protection against power spikes, surges, brownouts, and blackouts. You'll also gain maximum server uptime and decrease management costs. Award-winning features include:

- CellGuard™ intelligent battery management monitors battery performance and extends battery life.
- SmartSlot™ internal accessory slot lets you customize and enhance the performance of your Smart-UPS.
- QuickSwap™ user-replaceable batteries can be quickly and safely swapped out without powering down the connected equipment.



PowerChute *plus* provides unattended system shutdown and UPS management for Windows NT, Network and other servers. Manage Smart-UPS via SNMP, DMI and Web browsers (shown above). Features vary by operating system.

Plan for and control crisis situations

PowerChute *plus* FlexEvents™ lets you control UPSs reactions to power events. You can configure PowerChute *plus* to provide graceful, unattended server shutdown during an extended power outage or alert you to out-of-bounds environmental conditions before they result in costly downtime.

Web server and SNMP ready

APC's NEW WebAgent™ allows you to monitor and manage your Smart-UPS using your Web browser. New WebAlert™

notifies users of Web server shutdown via their browser. PowerChute *plus* also includes the PowerNet™ SNMP Agent

plug-in, which allows you to integrate your Smart-UPS with your existing SNMP management strategy.

Smart-UPS and PowerChute *plus* provide the complete solution in one convenient box. Server protection and peace of mind have never been easier.

Trade-UPS!

Fax or mail this coupon to APC and learn how you can easily trade in your old UPS for discounts towards a new Smart-UPS.

☐ **YES!** I'm interested in trading up an older competitors' or APC UPS to Smart-UPS. Please send Trade-UPS info.

☐ **NO** I'm not interested at this time but please send my FREE power protection handbook.

Name: _____

Title: _____

Company: _____

Address: _____

City/Town: _____

State: _____ Zip: _____ Country: _____

Phone: _____

How many servers on site? _____

Brand of UPS used? _____



Dept. A30

(888) 289-APCC x8116
Fax: (401) 788-2797
<http://www.apcc.com>

©1997 APC. All Trademarks are the property of their owners. SU01EF • (800)347-FAXX PowerFax • CompuServe: GO APCSUPPORT • E-mail: apctech@apcc.com • 132 Fairgrounds Road, West Kingston, RI 02892 USA

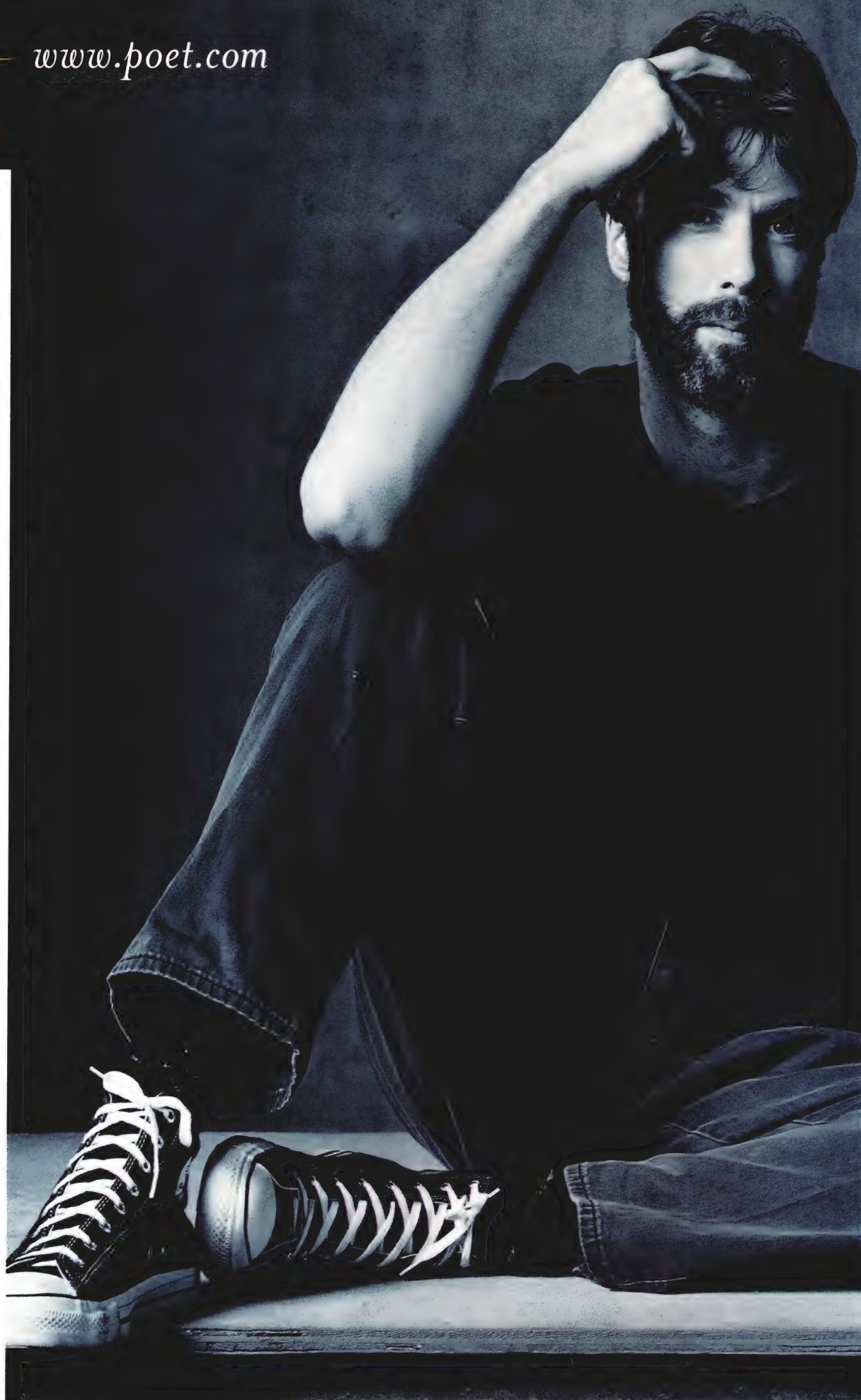
APC has won more awards for reliability than all other UPS vendors combined.



CIRCLE NO. 356 ON READER SERVICE CARD



www.poet.com



There once was a time when
objects meant
obstacles.

Now, it's all about
empowerment.

POET Software's easy-to-use object-oriented solutions
give developers new power over complex information.

And that lets you turn the design complication of intricate data into a
competitive advantage. At companies like Echelon Corp., Honeywell Inc.
and Novell, developers are using our advanced database technology to
put the most complex information to work in the most imaginative ways.

And now we're offering you the same opportunity. To find out how you
can use objects to bring it all under control, go to www.poet.com and
download a free SDK. Or call us directly at 1-800-950-8845.



The object is empowerment.

© 1997 POET Software Corporation. POET Software is a registered trademark.
All names are registered trademarks of their respective companies.

Motion Blur Effects

Greater realism through special effects

Tim Wittenburg

Most of us have taken pictures that ended up looking blurry. Usually, the cause of the blur is the subject moving through the camera's field of view, the camera shutter being set on a long exposure, the lens aperture not letting in enough light, or the film's ISO rating (film speed) being too low.

Photographers know that the rate of motion a film is capable of capturing without blur depends on a combination of factors within the camera, such as those just mentioned. The same phenomenon occurs in operating a video camera. The video circuitry, regardless of the underlying technology being used, also has an effective exposure. The human eye also has an effective exposure. The exposure characteristics of the sensor (a human eye or camera, for instance) place a limit on the rate of motion that can be captured by that sensor without blur. Stated simply, when an object's rate of motion exceeds the exposure limit of the sensor, blur results.

In contrast, computer-generated imagery techniques have no exposure limit associated with them. In effect, the computer's "camera" has an infinitely short exposure, while film, video, and the human eye do not. Consequently, rapidly moving objects generated by computers during special-effect sequences can appear

"too crisp" if motion blur is not properly added to the sequence. In this article, I'll present a method for blurring the appearance of moving objects. This method is applied after a sequence of images has been generated. The approach essentially involves temporal averaging over the sequence of output frames. That is, each pixel in the blurred output image is the average of a number of corresponding pixels from surrounding frames. The number of surrounding frames is called the



"blur depth." Smaller amounts of blurring can be produced by keeping the blur depth small, say, a value of 2 frames. Larger amounts of blurring can be produced by increasing the blur depth. Figure 1 illustrates how this method actually works. The sequence of images produced during animation are on the left side of Figure 1, labeled "Original Sequence," and numbered 1 through 7.

The blur depth indicates how many input frames are to be averaged for each blurred output frame in the sequence. In Figure 1, a blur depth of 2 is used. The illustration shows that a blurring window is passed over the sequence of input frames. Each blurred output frame is cal-

culated from 4 input frames. The first blurred frame is the average of four frames in the input sequence. The second blurred frame is the average of input frames 2 through 5. Figure 2 shows an example sequence with a blur depth of 4.

Using the method just described, you can create blurred sequences in which the amount of blur in each frame is an even multiple of the original frame rate. In other words, creating two blurred sequences in which the blur window is 4 and 6, respectively, creates sequences in which the amount of motion contributing to the blur is $4/30$ and $6/30$, respectively. This method is capable of varying the amount of motion contributing to blur only in whole increments of the original frame rate. Next, I'll present an enhancement that provides more flexibility over the amount of blurring that can be added to each frame in a sequence.

Controlling the Amount of Blur

The following technique provides greater flexibility in the amount of blur that can be added to computer-graphic imagery models that are composited into a special-effects scene. This additional control can be provided by oversampling the original sequence. Instead of producing 30 frames per second, the same amount of motion can be produced in, say, 150 frames per second—5 times the original rate. This means that each frame in the oversampled sequence represents $1/5$ of the motion contained in each frame of the original sequence. By varying the blur-window setting and applying the blurring operation to the oversampled sequence, each resulting blurred frame can be comprised of varying fractions of the original rate of motion.

Suppose, continuing with the example, that blurring is now applied to the oversampled sequence with a blur window of 7 frames. Each frame in the blurred sequence now contains $7/5$ the original rate of motion. What is needed next is a way to get back to the original frame rate of 30

Tim is a systems architect at the Star Tribune in Minneapolis and author of Visual Special Effects Toolkit in C++ (John Wiley & Sons). He can be contacted at 70403.3570@compuserve.com.



If you are currently

programming in

on a

platform,

then

is the

perfect editor for you.

Paste
program
language
stamp
here

Paste
operating
platform
stamp
here



Only Visual SlickEdit® can fill your every programming need, right out of the box. With it, you'll have superior power, speed and flexibility. Programming in any language, on any platform—even UNIX—is effortless. Deploy a single full-featured editor throughout your entire organization; and access an

unprecedented array of features, including SmartPaste™, API Apprentice™, a C/C++/Java code beautifier and difference editing. Plus, thanks to MicroEdge's 30 day risk-free trial, you can try it all for free. Call for your copy. And get ready to give Visual SlickEdit 2.0 your own personal stamp of approval.



VISUAL SLICKEDIT 2.0

THE MOST POWERFUL PROGRAMMER'S EDITOR ON THE PLANET. AND BEYOND.

CALL

1 800 934 EDIT (3348)

WWW

<http://www.slickedit.com>

VISUAL
SlickEdit

Visually Parse All Your Data

New! Version 2.1 with Javall

I highly recommend Visual Parse++
Larry O'Brien, Software Development

Visual Parse++ is the only tool that brings cutting edge parsing technology to all programmers. You can completely design any parser without ever leaving our sophisticated visual development environment. Whether you're new to parsing technology, or an experienced veteran, Visual Parse++ is the right tool for you.

Visual Parse++ provides native programming support for the following environments:

C/C++: full support for all compilers, including Visual C++ and Borland C++. Our class library is fully portable to all platforms, including Unix.

Visual Basic: use the powerful parsing ActiveX control

Delphi: native Delphi classes, mirroring our C++ support

Java - New! Visual Parse++ is now a must have with our new Java classes

Visual Parse++ also comes with drop-in parsers for **HTML, SQL, RTF, Java, C, C++,** and more. And don't listen to any yaccity-yacc from other products claiming to be visual. But don't take our word for it, download a free trial today!

www.sand-stone.com

Intro Price: \$299

Call 800-988-9023

Order your's today!



SandStone Technology Inc.
939 Coast Blvd, Suite 4C
La Jolla, CA 92037

Phone: 800-988-9023
Phone: 619-454-9404
Fax: 619-454-9467

<http://www.sand-stone.com>
e-mail: wild@sand-stone.com

(continued from page 36)

frames per second. The original frame rate can be achieved by undersampling the blurred sequence appropriately. If every fifth frame is now taken from the blurred sequence, the result is a sequence in which the rate of motion is again equal to the rate of motion in the original sequence—30 frames per second. However, the amount of blur in each frame corresponds to 7/5 the rate of motion in each frame of the original sequence. From this example, it is apparent that the rate of motion blur in each final frame relative to the original is a fraction in which the denominator is controlled by the oversampling rate and the numerator is controlled by the blur depth. Figure 3 illustrates these relationships.

Software Corner: Blurring Sequences

Listing One (listing begins on page 84) is the *motionBlur* function, which implements the blurring procedure in Figure 1. This function takes an image sequence as input and applies the blurring algorithm as previously described. A blurred output sequence of images is produced. The *firstImagePath* is a pathname to the first image in a sequence of images that is to be blurred. *OutputDir* is an absolute pathname to the output directory into which the blurred images are to be placed. The *numFrames* argument indicates the num-

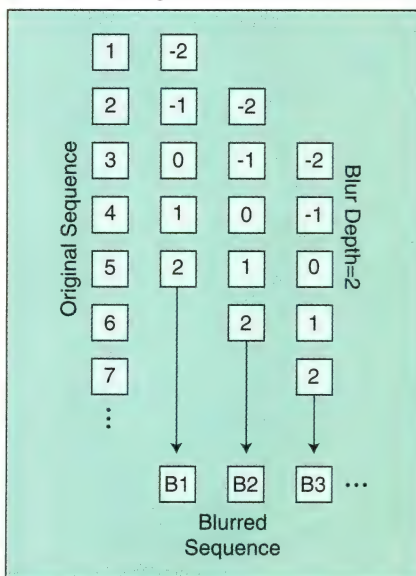


Figure 1: Motion blurring procedure.

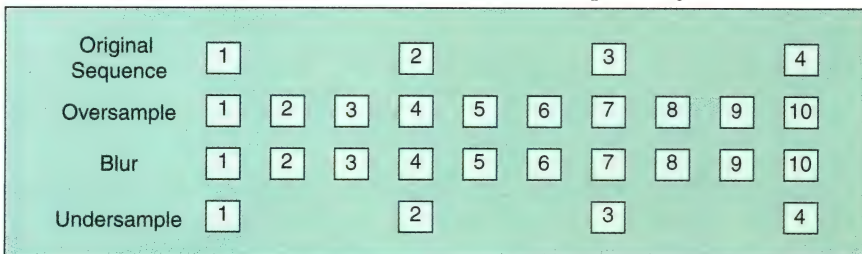


Figure 3: Controlling the amount of blur.

ber of frames in the image sequence to blur. The value of *blurDepth* specifies the number of images that are to participate in the blurring operation.

DDJ

(Listing begins on page 84.)

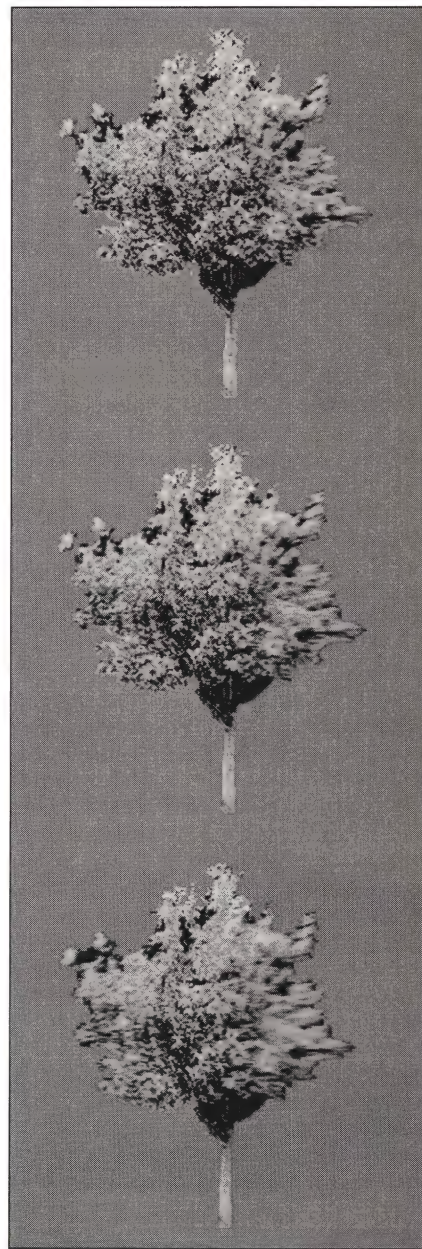


Figure 2: Sequence with motion blur, blur depth = 4 frames.



**Just what you'd expect from
an error detector 60 times faster than
the competition.**

A speedy way to get your hands on it.

As thousands of
developers know
firsthand, Purify® is the

leading tool for automatically

detecting execution errors and memory

leaks within complex software—even in DLLs and

third party code. In fact, Purify has benchmarked up

to 60 times faster than other products of its kind in



identifying run-time
errors in Windows NT®
and UNIX environments.

But don't just take our word

for it. For the fastest way to

improve the quality of your code, simply

download an evaluation copy from our website, call our

toll-free number or e-mail us at info-ddp@pureatria.com.



THE FOUNDATION FOR BETTER SOFTWARE

CIRCLE NO. 315 ON READER SERVICE CARD

©1997 Pure Atria Corp. Purify is a registered trademark and PURIFY'd is a trademark of Pure Software Inc. All other company names and products mentioned herein are trademarks of their respective companies. On August 26, 1996 Pure Software and Atria Software merged to form Pure Atria.

Treaps in Java

A simple and efficient general-purpose data structure

Stefan Nilsson

If you could use only one data structure, which one would you choose? A hash table? While it supports the basic insert, find, and remove operations, it doesn't keep the elements in sorted order. Therefore, it can't efficiently perform some tasks that are frequently encountered, such as finding the minimum element or producing an ordered list of all elements. The standard Java packages don't contain a data structure for ordered data.

What would you require of this ideal, sole structure? It should be easy to use (and preferably easy to implement); it should be able to hold an object of any class (as long as you provide a method for comparing objects); it should be efficient in terms of both speed and memory; and it should be thread safe.

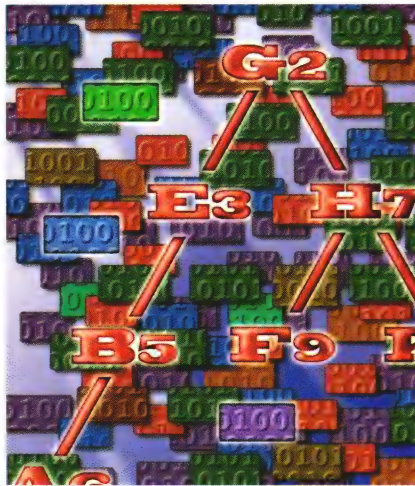
The randomized search tree ("treap"), devised by C.R. Aragon and R. Seidel and described in "Randomized Search Trees" (*Algorithmica*, 16(4/5):464-497, 1996), fulfills all of these requirements, and it offers the functionality of a general-purpose

Stefan is a lecturer in the department of Computer Science at the Helsinki University of Technology. He is doing research in the area of algorithms and data structures. Stefan can be contacted at Stefan.Nilsson@but.fi.

sorting routine, priority queue, hash table, stack, or standard queue.

The Data Structure

The data structure consists of a balanced tree. Therefore, an element is never more than about $\log_2 n$ steps away, where n is the number of elements in the treap and $\log_2 n$ is the binary logarithm (you may think of $\log_2 n$ as the number of bits needed to represent the number n in binary notation).



Being able to access the elements in the treap in $\log n$ time is a big improvement over some of the most common basic data structures; see Table 1.

The treap is reasonably space efficient as well. Each element is stored in a node in a tree, and each node contains an integer and two references, one to the left subtree and one to the right subtree. Hence, to store n elements in a treap, we need n integers and $2n$ references. The standard hash-table implementation in `java.util` uses n integers for the hash val-

ues and n references for the linked lists. Furthermore, it uses an array with a size that is typically close to n . For example, with a load factor of 1.0, the hash table will use the same amount of extra space as a treap if the table is full and more space if it's only partly filled. One problem with the `java.util` hash-table implementation is that the array doesn't shrink. Even if you remove all but one of the elements, the size of the array is still proportional to the maximum number of elements that were stored in the table during its lifetime. The treap doesn't have this problem.

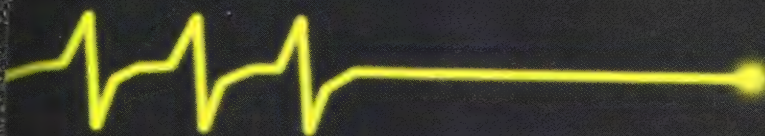
Interface

Java doesn't have a general method for comparing two elements. Therefore, you must make sure that an object that will act as a key in a treap implements a comparison method. To assure this, the treap package contains an interface, *Ordered*, that declares the abstract method `compareTo()` (see Listing One; listings begin on page 84). Any class that is to be used as a key in the treap must implement this interface.

To illustrate, I'll write a simple class, *Year*, that implements the interface *Ordered*. You are free to design the class as you see fit. The only requirement is that you implement the `compareTo()` method (see Listing Two). This is easy in this case: You just compare the integers representing the years. You need to use an explicit cast to tell the compiler that the argument is an instance of the class *Year*, because the comparison method implements the abstract method `compareTo()`, whose argument is of type *Ordered*.

With this out of the way, you can use the treap in the same way as a Java hash table. In this example, you use objects of

Do the technologies you work with look terminal?



Revive your systems career with hot, new technologies at Fidelity.

Every day, new developments in Internet/Intranet applications, information and telecommunications technology, and breakthrough real-time trading systems are being created at Fidelity by the systems experts of TempSource, Fidelity's in-house placement service for temporary systems assignments.

TempSource lets you work on systems developments that are revolutionizing the financial industry. And you'll use today's most sophisticated technologies — JAVA, UNIX, PowerBuilder, Visual C++, Windows NT — on your terms:

- Receive an hourly rate that's competitive with anyone in the industry
- Work on a short- or long-term basis
- Move from project to project or stay with one area of Fidelity
- Work in Boston, Marlborough, MA, or Merrimack, NH
- Have the security of a company with over \$450 billion in managed assets

So if you want to break boundaries, without being held back by antiquated technology, contact TempSource today.

Here are just a few of the technologies and positions available:

Developers for Web, Data Warehouse; PeopleSoft; Visual Basic; Java; UNIX; C++; Sybase and Oracle

MF, COBOL, CICS, DB2, IDMS, JCL Utilities, Year 2000 Developers

Quality Test Engineers

Release Engineers

Technical Writers

Positions available in Boston, MA, Marlborough, MA & Merrimack, NH. To apply, please email your resume indicating Job Code DA2183 to: tempsource@fidelity.com Or mail to: Fidelity Investments, Job Code WR2183, 82 Devonshire Street, Mail Zone A4C, Boston, MA 02109. Fax: 617-476-4264. Visit Fidelity's NEW Web site for more employment opportunities at www.fidelity.com/employment



**Fidelity
Investments®**

Fidelity Investments is proud to be an equal opportunity employer.

CIRCLE NO. 77 ON READER SERVICE CARD

Out-of-state candidates encouraged to apply.



(continued from page 40)

the class *Year* as keys, and strings as values. The call to the method *keys()* returns an enumeration of the treap. The enumeration is the standard Java enumeration class, whose only purpose is to return the keys of the treap in sorted order. Similarly, the method *elements()* returns an enumeration of the values.

Implementation

If you want to add an element to the middle of an array, you have to move a large number of other elements to make room for the new one. The most common way to avoid this problem is to use a linked list, where the elements don't reside at consecutive memory positions but are instead linked together: Each element in the list has a reference to the next one. In a

	Insert	Find	Min
Unsorted array	1	n	n
Sorted array	n	log n	1
Unsorted linked list	1	n	n
Sorted linked list	n	n	1
Hash table	1	1	n
Treap	log n	log n	log n

Table 1: Worst-case time for some basic operations in some basic data structures.

linked list, it's possible to insert an element in between two other elements just by changing two references. However, in a linked list, you can't directly access an element in the middle of the list, you have to follow a large number of links to find it. One standard technique to overcome this problem with arrays and linked lists is to use a search tree.

A search tree consists of a number of nodes. Each node contains a key and has two references, one to the left subtree and one to the right; see Figure 1. Note that the tree is ordered. All keys in the left subtree are smaller than the key in the node, and all keys in the right subtree are larger. This property holds for every node in the tree. Therefore, it's possible to search the tree in a simple manner. For example, to check if the key F is present in the tree, you start by comparing F to the key in topmost node (the root) of the tree. F is larger than E and you know that F must be in the right subtree (if it's in the tree at all).

Next compare F to H, the root of the right subtree. It's smaller, and you turn to the left, where you find the key you are looking for. You can use the same technique to insert a new node. If the key of the new node isn't already present in the tree, the search will take you to a posi-

tion at the bottom of the tree where the new node can be inserted.

But you still have a potential problem. The algorithm above is efficient only if the tree is well balanced. For example, if each node in the tree has only one child, the structure will behave just like a linked list: You have to visit every node in the tree to find the one at the bottom. There are plenty of algorithms available that try to keep a search tree balanced. Most of them are rather complicated. The treap offers a simple solution.

You will use rotations (the simplest possible rebalancing operation) to help keep the tree balanced. The idea is to change the form of the tree without disturbing the order of the keys. The left tree in Figure 2 is transformed into the right one by a "right rotation." This operation is easy to implement—only two references need to be updated, and the order of the keys is preserved. The corresponding balancing operation that transforms the right tree into the left one is called a "left rotation."

Now comes the stroke of genius. You want the tree to look as if it was constructed from a random sequence of updates. (As R. Sedgewick and P. Flajolet describe in *An Introduction to the Analysis of Algorithms*, Addison-Wesley, 1996, random trees are nicely balanced. The

VIREO UNCOVERS the SECRETS of DEVICE DRIVER DEVELOPMENT

Why are some developers able to build device drivers faster than others? What secrets do they know?

Vireo's Driver::Works and VTOOLS.D.

Driver::Works delivers on the power of NT and the promise of Vireo's award-winning **VTOOLS.D** products. **VTOOLS.D** is your complete solution for Windows 3.1 and Windows 95 device driver development. Both products provide:

- Faster, easier device driver development. Move quickly from Wizard to working driver.
- Multi-platform support.
- Full source code.
- Extensive real world examples.
- Complete support and service from the leaders in device driver technology.



For detailed product information, or to download the **FREE** Driver::Wizard, visit our Web site: **www.vireo.com**. Or call, 508-264-9200.

21 Half Moon Hill, Acton, MA 01720
Fax: 508-264-9205 Email: sales@vireo.com



Vireo Software

Driver::Works, Driver::Wizard, and VTOOLS.D are trademarks of Vireo Software, Inc. All other trademarks and registered trademarks are the properties of their respective companies.

expected length of an average search path is roughly $1.4 \log_2(n-2)$, and the expected length of the longest search path is about $4.3 \log_2 n$.) To achieve this, you assign a random priority number to each node before you insert it into the tree, and you make sure that each node in the tree has higher priority than its children. If you do this, the tree will have the same form as if the nodes had been added in priority order (that is random order).

The insertion algorithm is best explained by an example. In Figure 3(a) the node G (with priority 2) has just been added to the treap. It has been placed in the correct position with respect to the ordering of the keys, but it violates the priority requirement: It has higher priority than its parent. To fix this you perform a left rotation, Figure 3(b), to move it higher up in the tree. G still has higher priority than its parent and we perform a right rotation; see Figure 3(c). Finally, G is rotated up to the root of the tree; Figure 3(d). Now the tree fulfills both the key order and the priority order requirements.

At first this algorithm may seem hard to implement. It looks as if you would have to walk up the tree, which is hard since the references point in the other direction. However, this problem is easily overcome if you formulate the insertion algorithm recursively; see the pseudocode in Example 1.



Figure 1: A binary search tree.

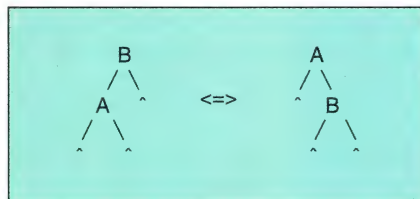


Figure 2: Rotation.

```

IF tree is empty RETURN node
IF node.key < tree.key
  Insert node into tree.left
  IF tree.prio > tree.left.prio
    rotate tree to the right
ELSE
  Insert node into tree.right
  IF tree.prio > tree.right.prio
    rotate tree to the left
RETURN tree
  
```

Example 1: Inserting a new node into a treap.

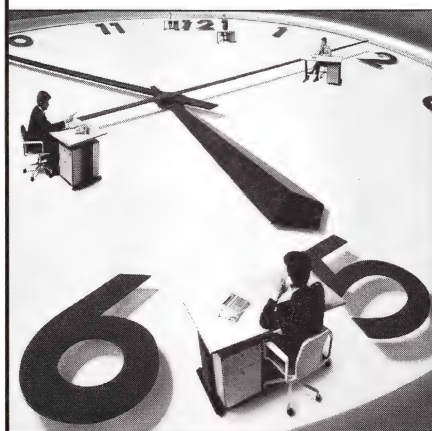
Compared to those of other balanced search tree schemes, the delete operation is also simple.

- If the node to be deleted is at the very bottom of the tree, you remove it directly and no rebalancing is necessary.
- If the node has only one nonempty subtree, you remove the node and replace it with the root of the subtree.
- If the node has two nonempty subtrees, you look at the priority numbers of the roots of the subtrees. If the left subtree has the highest priority, you rotate the root of this subtree to the left; otherwise, rotate the root of the other subtree to the right. This will move the node to be deleted further down the tree and

you have reduced the original problem to a smaller one that may be solved by calling the delete algorithm recursively.

The only piece of code in the Java treap implementation that is somewhat complicated is the enumeration—the object that returns the nodes in the tree in sorted order. A simple solution would be to have the enumeration store the latest key that was returned and find the next one by searching the tree starting at the root. To traverse the complete tree in this fashion would require at least $n \log n$ time. But it's possible to do it in linear time. The idea is to use an explicit stack to keep track not only of the current key but also the path from the root of the tree down to this

IT TAKES 6 TO 8 MONTHS TO INTEGRATE MOST IMAGING SOFTWARE. AND ABOUT A MINUTE TO LOSE A CLIENT WHO WON'T WAIT.



Need to upgrade the imaging function of your MIS system "yesterday?"

Samson Capture comes with the tools necessary to seamlessly integrate within legacy management systems in less than 12 weeks, not the 6 to 8 months others require.

What's more. Samson Capture is one of the only imaging applications that doesn't require your clients staff to learn new software or databases.

If you're interested in licensing for resale or image enabling your proprietary applications, call **1-800-582-0778** and ask about our Partner Alliance Program.

Samson Capture. It's the one suite of imaging applications with technological and marketing advantages built in.

SAMSON InfoTech™
Creative Solutions For Business™

Sales@mail.samsoninfotech.com

**Samson Capture™ Suite
Of Imaging Applications.**

**Seamless Integration
In Under 12 Weeks.**

CIRCLE NO. 73 ON READER SERVICE CARD

node. To find the next node, you continue down the tree to the right (pushing the nodes onto the stack). If there are no elements in the right subtree, you instead backtrack on the path using the stack. The code is complicated because you need to check whether you arrived at the current node from the left or from the right.

Finishing Touches

To turn the treap into a really useful general-purpose data structure, you should

add some more functionality, such as methods for reporting the size of the treap, removing the minimum (or maximum) element, and creating a clone or a string representation of the treap. You should also wrap up the classes and interfaces in a package and add Java style documentation. This is all straightforward pedestrian programming. The complete package is available electronically; see "Availability," page 3.

There are many ways to further optimize the code. Most of them will yield

only minor improvements. For example, eliminating the recursion from the insert and delete methods will make the code harder to maintain and the gain is likely to be only marginal, since simple function calls are efficiently implemented on most modern architectures.

The one optimization that is most likely to significantly improve the speed is to minimize the cost of the comparisons. A drastic way to do this is to use integers, rather than objects, as keys. The modification of the code is slight. All variables of type *Ordered* should be changed to *int*, and all calls to *compareTo()* should be replaced by an integer comparison. This modification will have to be done manually, since Java doesn't (yet) offer a way to write code that operates on both basic data types and objects.

Conclusion

The failure to use an appropriate data structure to maintain an ordered dynamic set of elements is one major reason why so many programs are slow. This problem can be solved by using the right data structure (and the treap is a very good choice).

DDJ

(Listings begin on page 84.)

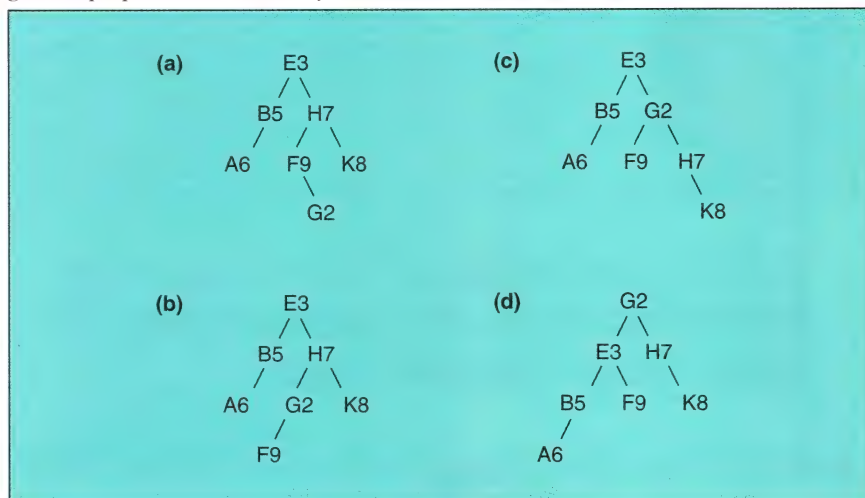


Figure 3: Inserting a new node G with priority 2 into a treap.

You're the best. We're the best. Let's get together.

CMG is a medium large software house, based in the Netherlands, the UK and Germany. CMG is well on its way becoming one of the largest independent IT companies in Europe.

The Advanced Technology division in the Netherlands is expanding rapidly and, as a consequence we are looking for more consultants.

We need the best project managers, the best technical project leaders, the best business consultants, the best infrastructure specialists and the best software engineers. In brief: we are looking for you. If you are a (Dutch-speaking) professional, then join the other 3,000 CMG people who are currently helping our customers expand their business, solve IT problems and are making a great career for themselves.

We work hard, we play hard and we pay above market-rates. In a nutshell, this is the most accurate way to describe how we succeeded as a profitable independent company for

over 30 years in the fastest growing marketplace in the world.

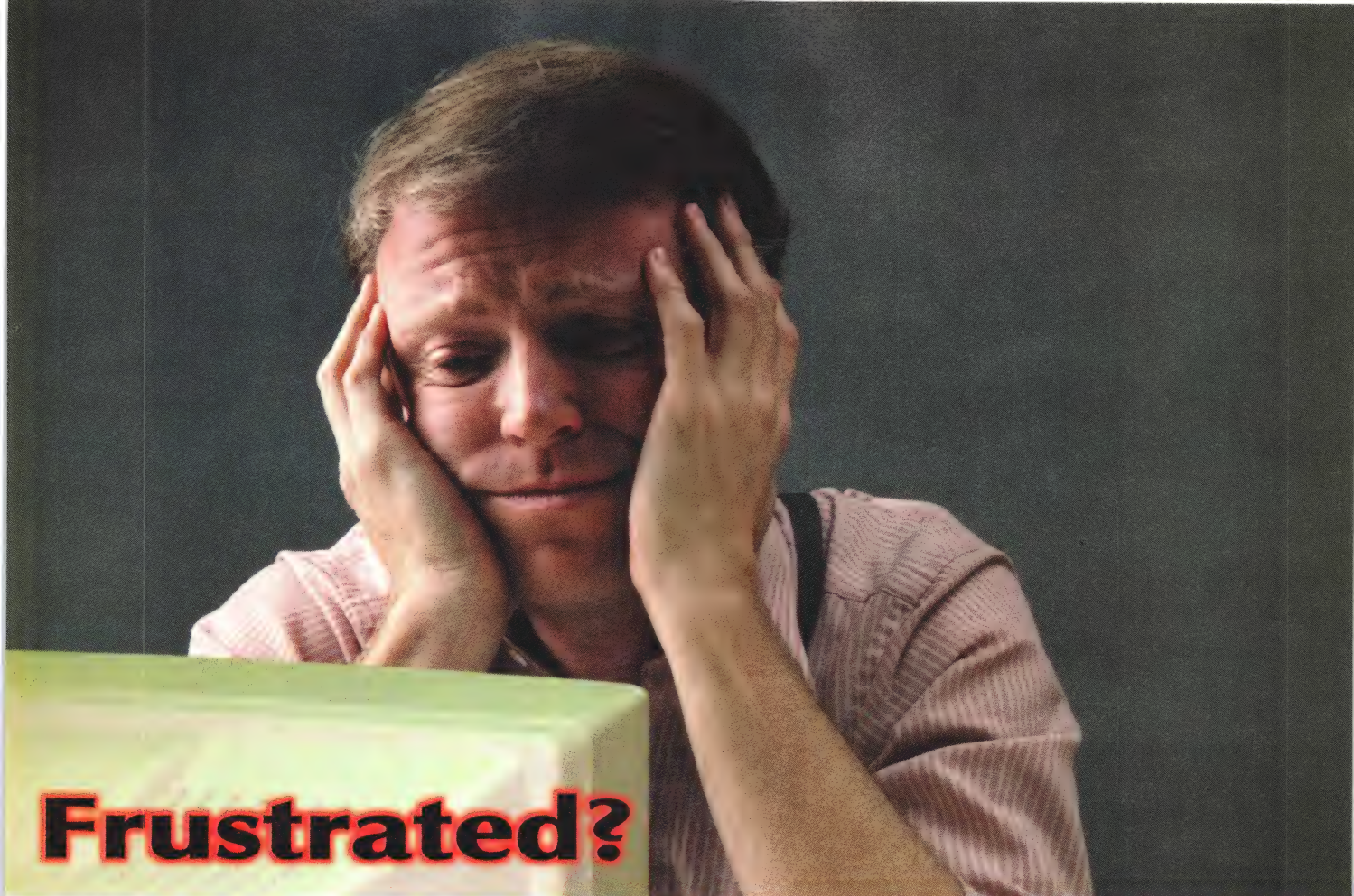
You're the best and we're the best to work for, according to an independent market analyst who publishes a yearly report called 'The best 25 IT-companies to work for'. In this list, CMG is mostly ranked first or second. Are you interested in what we do? Then visit our website, and form your own opinion.

Do you live in the Netherlands or are you planning to do so in the near future? Let us know and send your CV by mail, E-mail, fax or snail-mail, and we'll take it from there. Of course, you can also call us or use the recruitment form on the CMG homepage. CMG Finance, division Advanced Technology, P.O. Box 133, 1180 AC Amstelveen, The Netherlands. Rob Labordus, Associate Director.

Phone ++31 20 50 33 000, Fax ++31 20 50 33 011, E-mail rob.labordus@cmg.nl CMG's Internet address: <http://www.cmg.nl>

CMG
Information Technology

CIRCLE NO. 324 ON READER SERVICE CARD



Frustrated?

Your Debugger Is Not Enough!

**Announcing New
BoundsChecker 5.0
Visual C++ Edition**

Your deadlines are fast approaching. You're tired of the tedious, manual, and time-consuming task of debugging. And you can't find those killer bugs—API errors, memory errors, and resource leaks—that have eluded you, your team, and your existing debugging tools for days. It's time to get a solution that saves time, frustrations, and money, so you can deliver bullet-proof applications.

NuMega BoundsChecker 5.0 Visual C++ Edition. It's fast and automatic, so you can use it continuously and everyday to spend less time debugging! It's comprehensive, to find those killer bugs. It's completely integrated with Visual C++ 5.0, so you never have to leave the Developer Studio to check your applications. If you don't believe us... see for yourself. Download a free 14-day license from www.numega.com.

Find Bugs Faster... and Automatically!

BoundsChecker 5.0 Visual C++ Edition for Windows NT and Windows 95 ... \$499*
(for a limited time only)

Order Now! Call 1-800-4-NUMEGA (1-800-468-6342)

www.numega.com

Also available for Delphi, C++Builder, Borland C++, Watcom C++, Symantec C++, and 16-bit Windows.

*Per copy, 10 copy minimum.

NuMega Technologies Inc., the NuMega logo, Making Software Work, BoundsChecker, are trademarks of NuMega Technologies Inc. All other trademarks are the property of their respective owners. Copyright ©1997. All rights reserved.


NuMega
Technologies
Making Software Work

Atmel's AT89C2051 Microcontroller

A powerful 8051 by any other name...

Dhananjay V. Gadre

The Atmel AT89C2051 is a low-power, high-performance 8-bit microcontroller compatible with the MCS-51 instruction set and object code. The 20-pin AT89C2051, with its many hardware features, is especially attractive to 8051 developers because it is compatible with the 8051 and similar devices, and can reduce board space, components, and cost. In an application comprised of EPROM, address latch, an 8051, and other associated components, for instance, the 8051 offers enough on-chip digital I/O that additional external I/O components may not be required. In a similar AT89C2051-based system, however, a single AT89C2051 can replace the EPROM, latch, and 8051, providing the system software is limited to 2 KB.

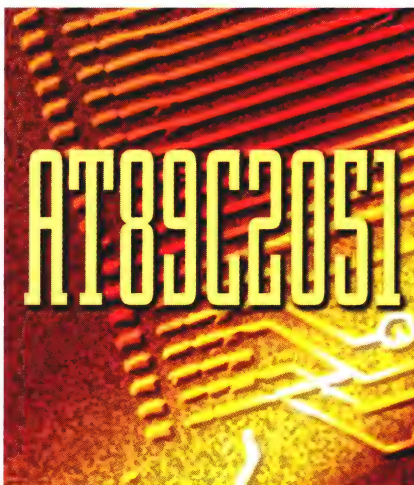
The AT89C2051 features:

- 2 KB of on-chip Flash program memory.
- 128 bytes of internal RAM.
- Fully static operation: 0 Hz to 24 MHz.
- Instruction compatible with MCS-51.
- 15 I/O lines.
- Full duplex programmable serial port.
- Two 16-bit programmable timers.

Dhananjay is a scientific officer for the instrumentation laboratory at the Inter-University Centre for Astronomy & Astrophysics (IUCAA) in Pune, India. He can be contacted at dvg@iucaa.ernet.in.

- On-chip analog comparator.
- Low-power and power-down modes.
- Wide operating voltages: 2.7 V to 6 V.
- 20-pin DIP/SOIC package.

The use of static memory allows the device to be operated at zero frequency. It also affords two software-selectable save-power modes. Idle mode stops the CPU, retaining the contents of the internal RAM, allowing the timer/counter, interrupt system, and serial ports to function normally.



Power-down mode saves the RAM contents but freezes the oscillator, disabling all other activity until the next hardware reset.

The amount of PEROM (programmable and erasable read-only memory) available on the AT89C2051 is sufficient for most applications, including use in portable instruments, supervisory-control applications, autonomous robots, and more. Use as controllers in portable instruments is further simplified by the low power consumption and wide operating voltage range.

The AT89C2051 allows 15 bits of I/O, configured as 8 bits on Port1 and 7 bits on Port3. Port1 and Port3 are compatible to the P1 and P3 on an 8051 (except Port1.0 and Port1.1).

Port1 pins P1.0 and P1.1 require external pullups. P1.0 and P1.1 pins also serve as inputs to an on-chip analog comparator (+ve and -ve inputs, respectively). Port1 output buffers have a 20 mA sink current capacity and can drive LEDs directly. By writing ones to the Port1 bits, they can be used as input bits.

As Table 1 shows, Port3 pins P3.0 to P3.5 and P3.6 are seven bidirectional I/O pins with internal pullups. P3.6 is internally connected to the output of the on-chip comparator and is not accessible as a general-purpose I/O pin. Port3 bits can also sink up to 20 mA of current, and when written with ones, can be used as inputs. Port3 pins also serve alternate functions (to be discussed shortly). If the user wants to use these alternate functions, the pin cannot be used for general-purpose I/O.

The AT89C2051 data sheet states that the on-chip oscillator can be used with a ceramic resonator as well as a resonant crystal element to provide the basic clock to the microcomputer. An external clock source with suitable levels can also be used instead of a crystal or a resonator. The operation is similar to that of an 8051. AT89C2051 can be operated with a clock frequency between 0 and 24 MHz. This is possible because the chip uses static memory.

Special Function Registers

The AT89C2051 has a register set identical to an 8051. Thus, it is possible to port existing 8051 applications to an AT89C2051 without change to the object code—as long as the software limits itself to the

From jet planes to Chunnel trains, QNX goes the distance: testing turbines, loading cargo, even controlling air traffic.

QNX helps healthcare applications run faster, and cost less, thanks to its executive-class speed and full x86 support.

Using a QNX-based vision system, shuttle astronauts stay focused on important things. Like launching satellites.

QNX serves up reliable service at the point of sale, whether it's handling fast-food orders or checking credit cards.



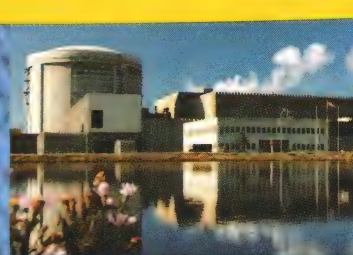
The **QNX** realtime operating system is found in these **worldwide** locations



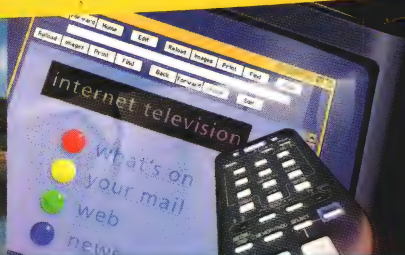
With its tiny, full-featured Internet suite, QNX helps web-transaction appliances do big business.



From traffic control to process control, QNX drives thousands of mission-critical applications nonstop, 24 hours a day.



When safety is measured in microseconds, nuclear power stations count on QNX: it's a *real* realtime OS.



Thanks to QNX, the web is coming to your living room faster than you can say "URL."

So many applications. So many demands. How does QNX do it?

Start with rock-solid OS technology field-tested for over 15 years. Add in innovative products like the award-winning Photon microGUI®, QNX's embeddable windowing system. Provide a rich, robust toolset so developers hit the ground running. And keep the memory footprint exceptionally small so runtime costs stay exceptionally low.

Most important, make it all fully scalable. That way, developers can deliver everything from web phones to factory-wide control systems—using a single OS.



The Leading Realtime OS for PCs

www.qnx.com

call 800 676-0566 ext. 1345

Available Now:

- internet toolkit
- embeddable GUI & browser
- POSIX and Win32 APIs
- embedded filesystems
- memory protection
- fault-tolerant networking
- distributed processing
- multilingual support
- unrivalled x86 support
- embedded OEM pricing

QNX Software Systems Ltd., Voice: 613 591-0931 Fax: 613 591-3579 Email: info@qnx.com

Outside North America: Voice: (44)(0)1923 284800 or 613 591-0931 Fax: (44)(0)1923 285868 Email: QNXeurope@qnx.com

© QNX Software Systems Ltd. 1997. QNX, Neutrino, and Photon microGUI are registered trademarks of QNX Software Systems Ltd. All other trademarks belong to their respective owners.

CIRCLE NO. 134 ON READER SERVICE CARD

**Desperately
seeking DOS for
fully compatible
LTR. Flexibility
and good memory
desired. Need
not be extra-
gant. Must be
good embedded.
Free demo ☎
1-800-221-6630**

**SWM, enjoys
computer games**

ROM-DOS™

Find your fully
compatible DOS
and save up to
80% compared to
MS-DOS®.
It even runs
Windows™.

That number again is
1-800-221-6630

Datalight®

System Software for Bright Ideas

18810 59th Ave. NE • Arlington, WA 98223
(360) 435-8086 • FAX: (360) 435-0253
e-mail: sales@datalight.com
www: <http://www.datalight.com>

ROM-DOS is a trademark of Datalight.
*MS-DOS and Windows are registered trademarks of Microsoft Corp.

CIRCLE NO. 151 ON READER SERVICE CARD

(continued from page 46)

available hardware resources, including memory and ports. This means that all jumps (*ljmp*) and calls (*lcall*) must be limited to maximum physical address 0x7FF. This also applies to all the other instructions that access memory in some form (for example, *cjne*, *jmp @A + DPTR*, *jnb*, and the like). Since the controller does not support external DATA or PROGRAM memory access, you should not use these instructions.

Programming the AT89C2051

Until you have a suitable "programmer"—a device similar to a PROM burner—using the AT89C2051 can be trickier than it would be with a simple 8031 or 8751. However, once a suitable programmer is available, using the microcontroller is straightforward. The AT89C2051 microcontroller can endure one thousand program and erase cycles. In this section, I'll present a simple programmer for AT89C2051 that is hosted on a 8052-based circuit running a Basic interpreter.

The AT89C2051 can be programmed using a suitable programmer out of the target

system. Table 2 identifies the various modes for erasing, programming, and verifying the chip. The code memory is programmed one byte at a time. After the controller has been programmed, to reprogram any nonblank byte, the entire chip has to be electrically erased. Erasing the chip is a simple task that takes a few milliseconds to execute.

The Atmel application sheets describe a simple programmer that lets you perform a variety of operations with the flash controller, including erasing, reading, programming, and verifying the contents of the target chip (see <http://www.atmel.com/atmel/products/prods10.html>). However, the driver software for the programmer does not support IntelHex object files. You can also purchase off-the-shelf programmers for these devices (see Airborn Electronics at <http://www.airborn.com.au/ab120.html>, for instance), or see if an existing programmer can be upgraded. I didn't have the time to build the circuit offered by Atmel, or to see if my existing EPROM/controller programmer could be upgraded to program the Atmel chips. Consequently, I decided to build a simple programmer using circuits I had available.

Since my lab already uses 8052-BASIC systems for various applications, I decided that I could add a little circuit to an existing 8052-BASIC evaluation board we built to make a simple (erase/program/verify) programmer. I found I could modify an existing IntelHex loader program for the Basic system to get the required programmer code. Figures 1 and 2 show the block diagram for programming and verifying the

Port Pin Alternate Function

P3.0	RXD (Serial input port)
P3.1	TXD (Serial output port)
P3.2	NT0 (External interrupt 0)
P3.3	INT1 (External interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)

Table 1: Port3 alternate functions.

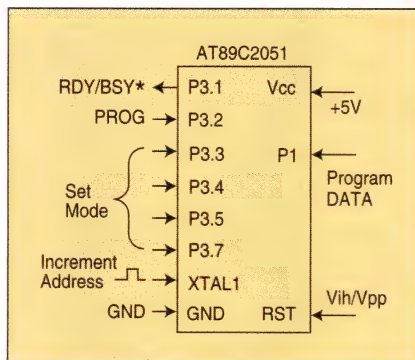


Figure 1: Programming the Flash memory in AT89C2051.

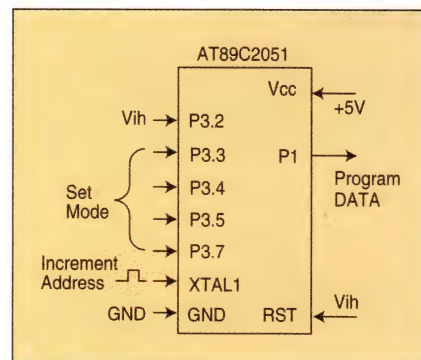


Figure 2: Verifying the Flash memory in AT89C2051.

Mode	RST	P3.2	P3.3	P3.4	P3.5	P3.6
Write Code Data (1,3)	12V	Low Pulse	L	H	H	H
Read Code Data (1)	H	H	L	L	H	H
Write Lock Bit 1	12V	Low Pulse	H	H	H	H
Write Lock Bit 2	12V	Low Pulse	H	H	L	L
Chip Erase	12V	Low Pulse(2)	H	L	L	L
Read Signature Byte	H	H	L	L	L	L

Table 2: AT89C2051 programming modes. (Note: Internal PEROM address is reset to 000H on the rising edge of RST and advanced by a positive pulse on XTAL1 pin. Chip erase requires a 10 mS pulse. P3.1 is pulled low during programming.)

Flash-memory contents of an AT89C2051. Figure 3 shows the circuit schematic for the programmer. The programmer needs a suitable +5V power supply for the 8052 and +12V supply for the programmer to generate Vpp. The +12V power supply must be stable and provide a voltage between 11.75V and 12.5V. The limits for the programming voltage for the AT89C2051 are between 11.5V and 12.5V. A suitable 20-pin ZIF socket can be used to insert and remove the AT89C2051 chip. Listing One (listings begin on page 55) is the Basic program for programming the AT89C2051. A PC with an 8051 assembler and terminal-emulation program (I use Vterm) are the only tools required for programming and using the AT89C2051.

To find out if my programmer was correctly burning the code into the AT89C2051, I used the test code in Listing Two. After I was satisfied that things were working, I went ahead with more serious applications.

Using the Programmer

To use the programmer, the ZIF socket is loaded with the AT89C2051 to be programmed before power is applied to the system. This is followed by applying power (+5V as well as +12V) to the circuit. The Vpp generation circuit applies a log-

ic 0 to the RST pin at power on. Subsequently the 8052 system is loaded with the code in Listing One from within the terminal-emulator environment. To communicate with the 8052-BASIC system, the terminal emulator can use any standard baud rate as well as transmission

*The amount of
PEROM available is
sufficient for many
applications*

settings (number of bits, parity, and so on). However, the 8052-BASIC requires that, after power on, the host sends a SPACE character as the first character. This allows 8052-BASIC to adjust to the baud rate and other transmission settings. After getting the READY prompt, Listing

One is downloaded and executed. The Basic program is then ready to accept IntelHex format object files. The received hex code is stored in the External RAM of the 8052 system. After the hex file is successfully received, the programmer proceeds with erasing the AT89C2051, then begins programming it with the user object code. After the code is burnt into the chip, the programmer power supply is turned off before removing the target chip. The target chip is then ready to be inserted into the target system.

Using the AT89C2051

A good use for this controller is in any application that needs to keep current consumption low, with features like power-down, or sleep modes of operation, and that have a serial port, timers, interrupts, and I/O pins.

For instance, I needed a 12-bit multi-channel ADC to connect to PCs in a certain application. Rather than invest in new parallel ADCs, I decided to see if our existing inventory of MAX186 ADCs could be of any use. MAX186 had everything we needed, except that the chip operates at serial clock with a minimum clock-frequency requirement of 100 KHz (which would be difficult to generate under program control on older PCs). Consequently,

What'ssss eating you?

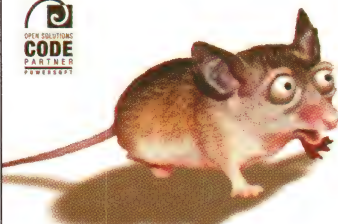
Lost in a jungle of languages? Crushed in the coils of immense binary files?
Unable to escape the voracious jaws of impending deadlines? Are your
development projects *eating you alive*?



**multi
edit**

OK, it's probably not *quite* that bad. Still, you are confronted daily with the challenge of managing large numbers of source files, using multiple languages & compilers, integrating with different vendors' tools, and coordinating with other developers' work. The last thing you need is to be burdened with inflexible, under-powered editing tools. You need an editor designed to adapt to your own individual work style, that supports the languages you work with, one that will **SAVE YOU TIME!** Multi-Edit is your answer.

For more than eight years, Multi-Edit has increased programming productivity world-wide. With extensive support for C/C++, Delphi, Java, Fortran, COBOL, HTML, PERL (just to name a few), and features like code templates, hex-editing, file compare, VCS integration, multi-file search and replace, and more, Multi-Edit will do the same for you!



"Unbelievable, the best editor I've ever used..."
-- Ross Mason; Shelby Systems, Inc.

Get an in-depth look at what Multi-Edit can do for you.
Visit our web site and check out our fully functional demo!

www.amcyber.com
1-800-899-0100



1830 West University Dr. Ste. 112
Tempe, AZ, 85281
Voice: (602) 968-1945
Fax: (602) 966-1654

Suggested
Retail Price
\$199





Multi-Edit is a trademark of American Cybernetics, Inc. Other products are trademarks of their respective publishers. Delphi is a trademark of Borland. Developer tested only. American Cybernetics stands behind all of its products with a 30 day Money Back Guarantee. No animals were eaten by other animals during the making of this ad.

CIRCLE NO. 143 ON READER SERVICE CARD

I decided to build a general-purpose interface that could be used in other applications. I found that the AT89C2051 worked well. The result of my design was a solution that offers a nibble wide input and output interface that, though tailored to connect to the PC parallel port, could be used anywhere else.

Using an 8051 capacity controller to parallelize serial ADC data might be considered overkill. While there are parallel output 12-bit multichannel ADCs of similar performance, I decided to use available components to get the system working. The AT89C2051 uses its serial port signal pins TxD and RxD to connect to

the MAX186. The microcontroller serial port operates in mode 0, in which the serial port works as a shift register, either as input or output. In the shift register mode, the TxD pin supplies the shift clock, and the RxD pin provides the data or reads the external data as per the direction. The controller programs the serial port as an output shift register in the beginning of the acquisition cycle during which the MAX186 needs the 8-bit control byte that contains conversion parameters, channel number, and so on. After the 8-bit data is shifted out, the controller program converts the serial port as an input shift register and reads back the converted ADC data as 2 bytes. Figure 4 shows the block diagram, and Figure 5 (see page 55) shows the circuit schematic. Figure 6 shows the wiring scheme to connect the controller board to the PC printer adapter.

Mode	Action
0	Start ADC conversion on channel 0
1	Start ADC conversion on channel 1
2	Start ADC conversion on channel 2
3	Start ADC conversion on channel 3
4	Start ADC conversion on channel 4
5	Start ADC conversion on channel 5
6	Start ADC conversion on channel 6
7	Start ADC conversion on channel 7
8	Read nibble 0 of the last ADC conversion result
9	Read nibble 1 of the last ADC conversion result
10	Read nibble 2 of the last ADC conversion result

Table 3: Modes of operation.

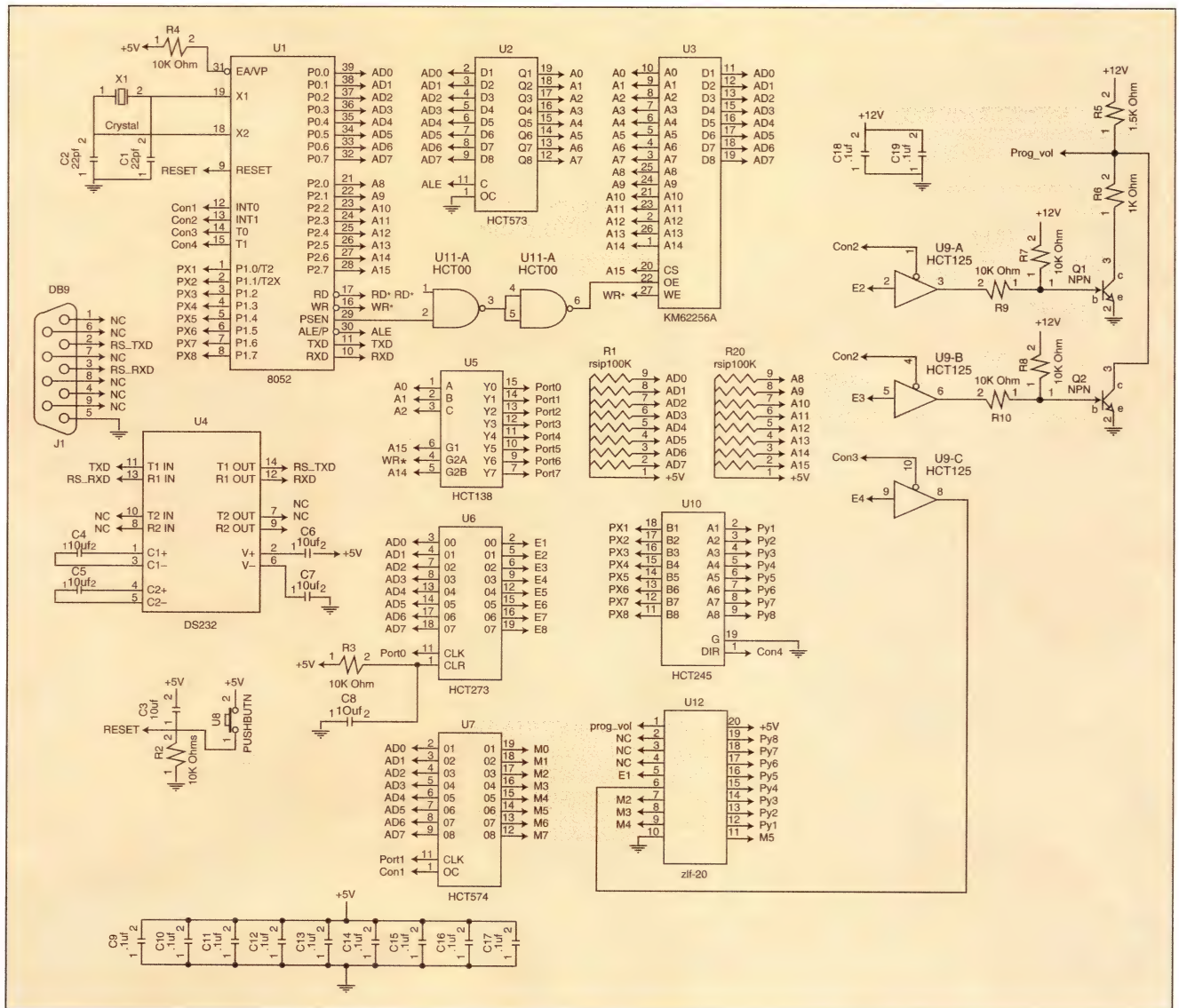


Figure 3: Programmer for the AT89C2051. All resistors are 5 percent CFR, except R5 and R6, which are 1 percent MFR. Transistors are BC547.

THE **LEADER** IN EMBEDDED INTERNET PROTOCOLS

TCP/IP

**Web Management
SURF360/386**

100% RFC Compliant

In FUSION

SNMP

OSPF

RIP

PPP

Powerful internet Protocols Made Easy!



Pacific Softworks

Welcome to the FUSION Profiler.

To set up FUSION for your application, choose set up from the menu above, then select a level option.

Choose the Compiler you will be using.

- ☒ Microsoft
- ☐ Borland
- ☐ Microtec
- ☐ SDS Crosscode
- ☐ MetaWare

Previous

FUSION Profiler - Advanced - OS Selection

Choose the Operating System you will be using.

- ☒ FUSION OS
- ☐ AMX
- ☐ pSOS
- ☐ Nucleus
- ☐ Microtec VTRX
- ☐ Phar Lap TNT

Previous

☐ Wind River

Choose the Processor you will be using.

- | | |
|---------------------------------------|--|
| ☐ AMD 290XX | ☐ ARM |
| ☐ Intel 80X86 | ☐ Motorola 68XXX |
| ☐ SGS Thompson | ☐ TI TMS 320 Family |
| ☐ MIPS | ☐ Fujitsu |
| ☐ NEC | ☐ Toshiba |
| ☐ Hitachi | ☒ Other |

Previous

Next

NEW!

FUSION Profiler

FUSION, the most powerful Internet protocols speeds your time-to-market and saves you \$\$\$\$ in development costs!

FUSION Profiler gives you point-and-click control over your TCP/IP development for Embedded systems. Just select your environment, and FUSION Profiler walks you step-by-step through your configuration. This means: ease of configuration, and a perfect build every time.

Using the FUSION Profiler enables almost anyone to configure these powerful FUSION applications for embedded systems.



Pacific Softworks

FOR MORE INFORMATION CALL:

(800) 541-9508 within USA

(805) 484-2128 outside USA

(805) 484-3929 FAX

Internet: sales@pacificsw.com

Europe, Middle East and Africa

Ph: (44) 1494-432-735

FAX: (44) 1494-432-728

World Wide Web: <http://www.pacificsw.com>

Distributors Worldwide

CIRCLE NO. 340 ON READER SERVICE CARD

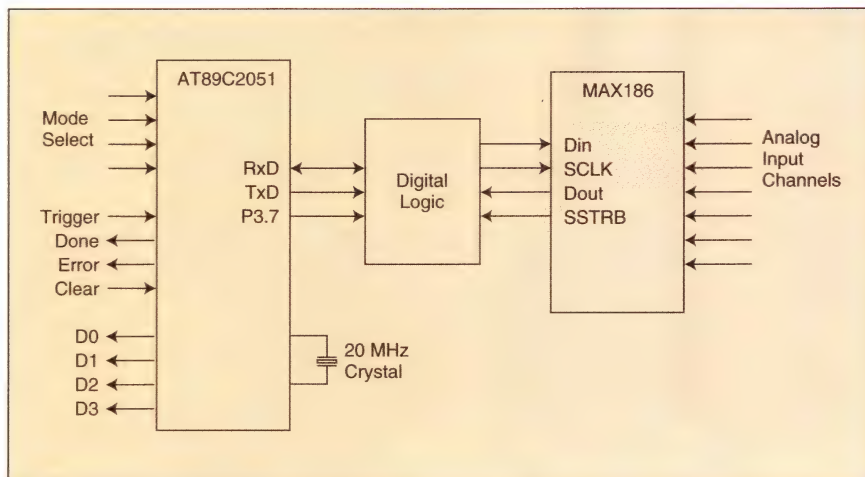


Figure 4: Connecting the AT89C2051 to serial ADC.

(continued from page 50)

The user interface of the converter consists of the following signals:

- 4 bits of mode inputs that determine the mode of operation for the converter.
- A trigger input that triggers the converter into the requested mode.
- A clear-status input that is used to erase previous status information.

The converter outputs are:

- 4 bits of data.
- A done flag that indicates the end of operation.
- An error flag indicating an attempt to launch a nonimplemented mode of operation.

The mode input to the converter determines what task the controller will

perform when it is triggered. With 4 bits of mode input, up to 16 modes (see Table 3) of operation can be implemented. For this design, only 11 combinations are required; the rest can be used later for expansion.

Using the Converter

The converter interface is designed so that it can be used in any embedded application. The interface is ideally suited for data acquisition on PC compatibles using the parallel printer adapter signals. The converter provides access to eight channels of 12-bit ADC. The analog input voltage range of the ADC is 0 to 4.095 volts and at 12 bits, a resolution of 1mV.

Listing Three is C code to interface the controller through the PC parallel port.

Conclusion

I first heard about Atmel microcontrollers from Russ Hersh's 8051FAQ. I thank Tina Anagnos, David Lee, and Jim Tatsukawa at Atmel Corp. (<http://www.atmel.com/>) for providing data sheets, answering my questions regarding the programmer, and for providing AT89C2051 samples. My good friend Dr. Pramod Upadhyay needed a 12-bit AD converter for PCs just when I was looking for an application.

DDJ

(Listings begin on page 55.)

NEW!

Dr. Dobb's

Alternative Programming Languages CD-ROM, Volume 2.0

Price \$49.95

Due to popular demand, Dr. Dobb's has collected another volume of programming languages and tools. The editorial staff of Dr. Dobb's has once again assembled the most useful, interesting, and essential programming languages available and created *The Alternative Programming Languages CD-ROM, Volume 2.0*. This newest release contains over 35 programming languages, plus a large number of tools and utilities all on this one CD-ROM.

Whether involved in object-oriented development, internet development, text processing or document formatting, this definitive collection of languages and tools will help any serious programmer increase their productivity and reach that next level of programming excellence.

Here's what you get...

- Java • Modula3 • Oberon 2 • Smalltalk X • Sather
- Eiffel • Small Eiffel • Haskell/GOFER • ML • J • DSC
- uc/C++ • Glisch • Strand88 • Parasol • Phi (Phantom)
- Cocktail • Python • Tcl • Bob • Perl • Bob DLL • ReXX
- Netrexx • Lout • Ghostscript • Slang • Lcc • Gnu C
- CSC • Modula 2 • G77 • Pascal • Dylan • Forth • Neudl
- Quincy • D-Flat • D-Flat++ • And more!

These powerful and innovative languages come with source code for compilers or interpreters, enabling you to put this CD-ROM to work immediately on your programming projects. This is truly the most extensive collection of programming languages ever assembled.

Visit our web site for a more detailed explanation - www.ddj.com/cdrom/

ORDER TODAY!
U.S. & Canada:
800-992-0549
All Other Countries:
913-841-1631

E-mail: orders@mfi.com
Fax Orders: **913-841-2624**
Mail Orders:
Dr. Dobb's CD-ROM Library
1601 West 23rd St., Ste. 200
Lawrence, KS 66046-2700 USA

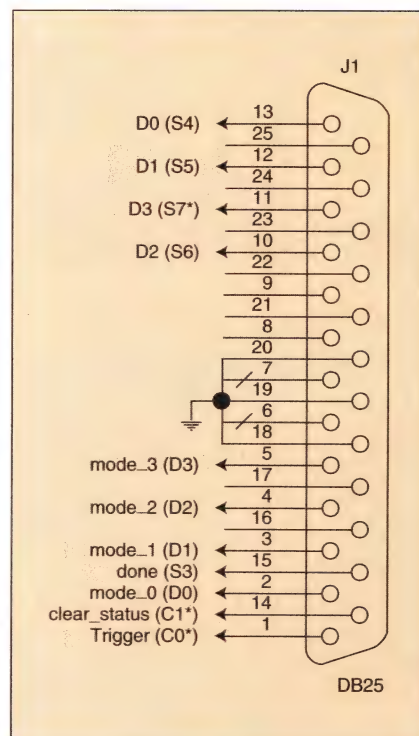
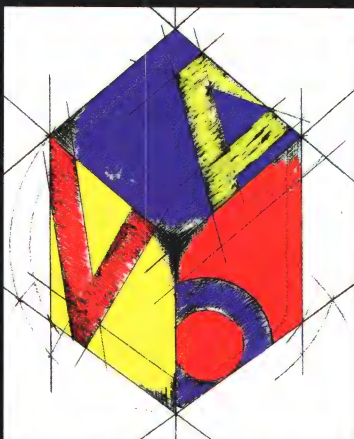


Figure 6: Wiring diagram for connecting the interface to the printer port.

the COMPLETE PACKAGE



SOFTWARE

- C Cross Compilers
- Macro Assemblers
- Simulator Debuggers
- Integrated Development Environments (IDEs)
- Real Time Operating System (RTOS)

HARDWARE

- In-Circuit Emulators
- Eeprom Programmers

CHIP SUPPORT

- 8051
- 80C51XA
- 68XXX FAMILY
- Z80/Z180/64180
- 68HC05
- 68HC11
- 8096/80196
- PIC 16CXX/PIC 17CXX
- 6800/6801/6803
- 6500/6502

and Much More

AVOCET

SYSTEMS®, INC.

120 Union Street • P.O. Box 490
Rockport, Maine 04856 USA
avocet@midcoast.com
http://www.midcoast.com/~avocet
FAX: (207) 236-6713

(800) 448-8500
(207) 236-9055

IC2000 - ICE

With the introduction of the iC2000 development system a new era of emulation technique has begun. Using the latest FPGA technology an extremely flexible real-time in-circuit-emulator has been developed, which supports more than 160 derivatives of popular 8-bit and 16-bit micro controllers.

The iC2000 is completely configurable by software. Only the POD, which connects the emulator to the target system, has to be changed to switch from one MCU to another or to add support for new micro controllers. The iC2000 already supports a wide range of MCUs and new supported devices are being added every week.

Processors Supported

Z80/180

8031/51

68HC11

68HC05

PIC 16/17

6809

80C186EM/ES

80C18x

68HC16

80C196

USB 82930A

68HC3xx

V25



Features

- Debug without halting your target system
- Performance analyzer and code coverage monitor
- Background Interrupt Mode
- Background DMA Mode for Real Time Watches
- 3 Channel Pattern Generator
- Connects to Serial or Parallel Port
- Fully compatible and integrated with Avocet Systems as well as other software tools.
- iC2000 base unit with emulation module will take PODs for all major 8 & 16 bit chips
- Powerful trace & logic analyzer options
- Complex break conditions

iSYSTEM

THE TOOL COMPANY
iSYSTEM USA, a division of Avocet Systems, Inc.

CIRCLE NO. 282 ON READER SERVICE CARD

Visit us on the web:

www.midcoast.com/~avocet



Don't Limit Your Product's Capabilities—Use Web Technology.

Visit us at <http://smallest.pharlap.com> to see how you can develop embedded applications that take full advantage of the web.

Making The Most Out Of Your Products

Phar Lap's TNT Embedded ToolSuite[®], Realtime Edition, complete with Realtime ETS[™] Kernel, now comes with our robust networking protocol—ETS TCP/IP! This cutting-edge feature allows your customers' mainframes, workstations, or PCs to communicate with products on the factory floor, in the lab, or at remote sites—all using Web technology!

As a result, electronic OEMs can use this technology to create intelligent machines

and instruments for an unlimited number of applications such as medical instruments, robotics, avionics equipment etc.

In addition, we support a variety of network protocols as well as industry standard tools such as Visual C++, Borland C++, CodeView, and Turbo Debugger.

Never Underestimate The Power Of The Web

Because Phar Lap's TNT Embedded ToolSuite comes with development tools, a Realtime ETS Kernel and ETS TCP/IP, it is a one-stop shopping product with enormous power!

To find out more, catch us on the "World's Smallest Web Server" today at URL <http://smallest.pharlap.com> or call a Phar Lap sales representative. Let your products realize the power of Web technology!

The 32-Bit x86 Experts



Embedded Development — Simply on Target[™]

Phar Lap Software, Inc. 60 Aberdeen Avenue, Cambridge, MA 02138 • Tel: (617) 661-1510 • Fax: (617) 876-2972 • <http://www.pharlap.com>

CIRCLE NO. 99 ON READER SERVICE CARD

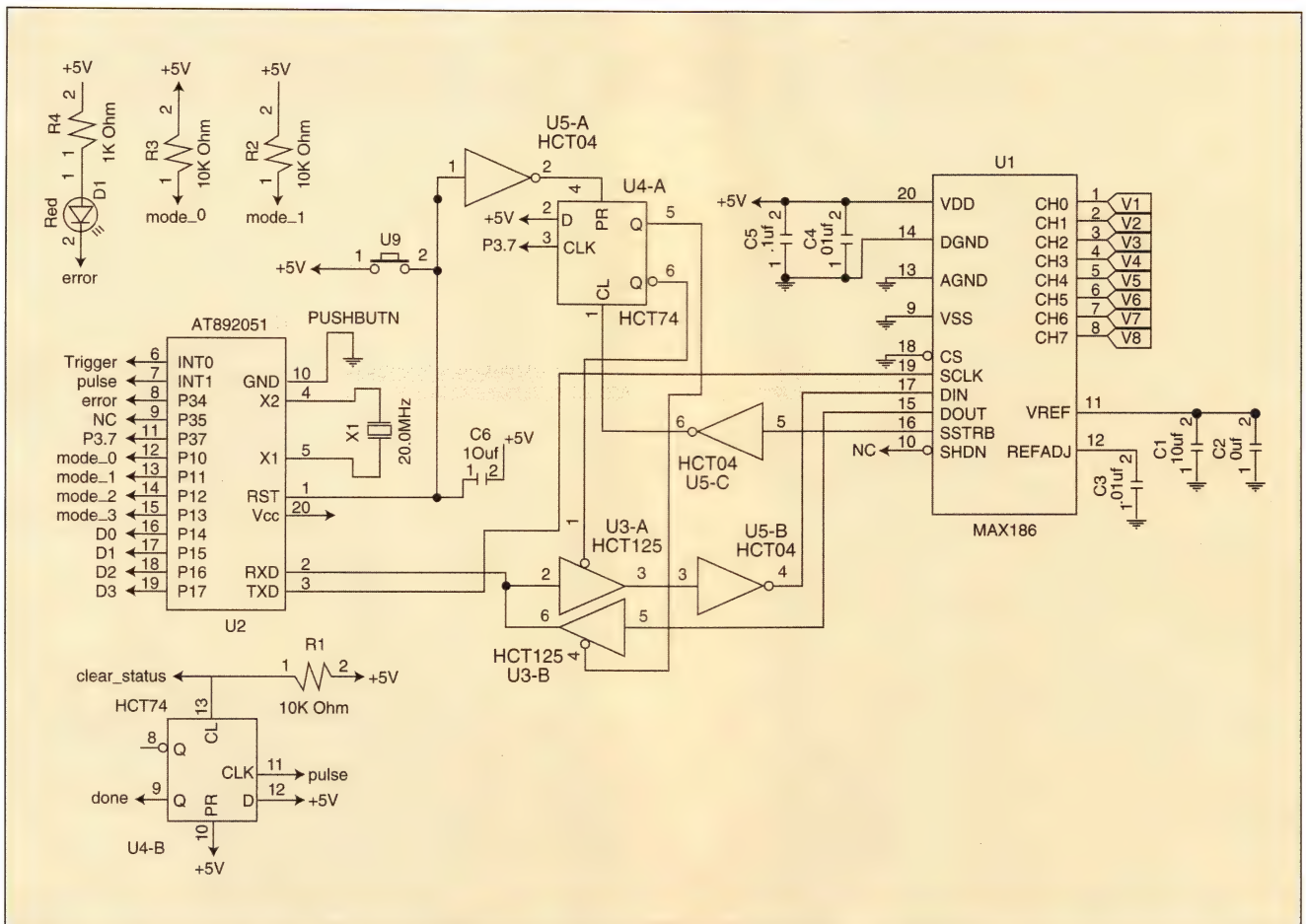


Figure 5: Circuit schematic for the AT89C2051 and MAX186 interface.

EMBEDDED SYSTEMS

Listing One

```

1 REM Programmer for ATMEL AT89C2051
2 Print "Atmel AT89C2051 Programmer"
3 Print " Accepts IntelHex files "
4 Print "   Dhananjay V. Gadre   "
5 Print "       1996       "
6 MTOP=41FFH
10 A=GET : IF A=0 THEN 10
20 IF A<> 58 THEN 10
30 PRINT CHR(A),
35 CSUM=0
40 GOSUB 1000
50 LL=A : IF A=0 THEN 230
55 CSUM=CSUM+LL
60 GOSUB 1000
70 ADDR=256*A
75 CSUM=CSUM+A
80 GOSUB 1000
90 ADDR=A+ADDR: IF ADDR > 1FFFF THEN GOTO 4000
100 CSUM=CSUM+A
110 GOSUB 1000: IF A<> 0 THEN 230
130 FOR T=1 TO LL
140 GOSUB 1000
150 CSUM=CSUM+A
160 XBY(ADDR + 4200H)=A
165 D=CBY(ADDR + 4200H): IF A<>D THEN PRINT "ERROR"
167 IF A<>D THEN GOTO 3000
170 ADDR=ADDR+1
180 NEXT T
190 GOSUB 1000
200 CSUM=CSUM+A : CSUM=CSUM.AND. 00FFH
210 IF CSUM<>0 THEN 3000
220 PRINT : GOTO 10
230 GOTO 2000
1000 A=GET: IF A=0 THEN 1000
1010 PRINT CHR(A),:IF A<58 THEN A=A-48: GOTO 1030
1020 A=A-55
1030 C=A
1040 A=GET: IF A=0 THEN 1040
1050 PRINT CHR(A),:IF A<58 THEN A=A-48: GOTO 1070
1060 A=A-55
1070 A=16*C+A

```

```

1080 RETURN
2000 MXAD=8001H: EXAD=8000H
2010 XBY(4C00H)=00C2H: XBY(4C01H)=00B2H: XBY(4C02H)=0022H
2020 XBY(4C03H)=00D2H: XBY(4C04H)=00B2H: XBY(4C05H)=0022H
2030 XBY(4C06H)=00C2H: XBY(4C07H)=00B3H: XBY(4C08H)=0022H
2040 XBY(4C09H)=00D2H: XBY(4C0aH)=00B3H: XBY(4C0bH)=0022H
2050 XBY(4C0cH)=00C2H: XBY(4C0dH)=00B4H: XBY(4C0eH)=0022H
2060 XBY(4C0fH)=00D2H: XBY(4C10H)=00B4H: XBY(4C11H)=0022H
2070 XBY(4C12H)=00C2H: XBY(4C13H)=00B5H: XBY(4C14H)=0022H
2080 XBY(4C15H)=00D2H: XBY(4C16H)=00B5H: XBY(4C17H)=0022H
2090 PRINT "Erasing AT89C2051...": CLOCK1: PRINT TIME
2100 rem RST=H, P3.2=H
2110 EX=000Ah: XBY(EXAD)=EX: call 4c06h: call 4c0ch
2115 REM Wait 10 ms
2120 For t=0 to 5: next t
2125 REM Set P3 for Erase
2130 MX=0004h: xby(MXAD)=MX: call 4c00h: FOR T=0 TO 5 : NEXT T
2135 REM Set RST to 12V
2140 EX= EX .AND. 00FDH: XBY(EXAD)=EX: FOR T=0 TO 5: NEXT T
2145 REM Set P3.2 to 0
2150 EX=EX .AND. 00F7H: XBY(EXAD)=EX: FOR T=0 TO 5 : NEXT T
2155 REM Set P3.2 to 1
2160 EX=EX .OR. 0008H: XBY(EXAD)=EX
2165 REM Put RST to 5V
2170 EX=EX .OR. 0002H: XBY(EXAD)=EX
2180 PRINT "AT89C2051 Erased..."
2190 PRINT "Writing and Verifying..."
2200 FOR COUNT=0 TO 07FFH
2210 VAL1=XBY(4200H+COUNT)
2215 REM Set P3 to WRITE data
2220 MX=0038H: XBY(MXAD)=MX
2225 REM Put data on P1
2230 PORT1=VAL1: CALL 4C12H
2235 REM Put RST to 12V
2240 EX=EX .AND. 00F9H: XBY(EXAD)=EX
2245 REM Put P3.2 to 0
2250 EX=EX .AND. 00F7H: XBY(EXAD)=EX
2260 FOR T=0 TO 5: NEXT T
2265 REM Put P3.2 to 1
2270 EX=EX .OR. 0008H: XBY(EXAD)=EX
2275 REM Disable data from source
2280 CALL 4C15H
2282 EX=EX .OR. 0002H: XBY(EXAD)=EX
2285 REM Set P3 to read data
2290 MX=30H: XBY(MXAD)=MX

```

(continued on page 59)

Serialize.h++

- Allows you to modify only your C++ apps; no need to change your Java apps
- Offers a way to exchange data and objects easily between C++ apps and any Java apps that use the Java serialization mechanism
- Includes pre-built ReaderWriter objects that map popular Tools.h++ classes to corresponding classes in the Java core libraries

JTools 2.0

- Allows exchange of data and objects between Java apps and any C++ app that uses Tools.h++ virtual streams
- Provides virtual streams for Java classes that correspond to the Tools.h++ virtual stream mechanism
- Includes pre-built Streamers that map Tools.h++ classes to corresponding Java core library classes or to JTools classes

Tools.h++

- Packed with over 120 reusable foundation-level classes for all your C++ development needs
- Includes virtual streams for persistence
- Provides a higher-level interface to the Standard C++ Library

Standard C++ Library

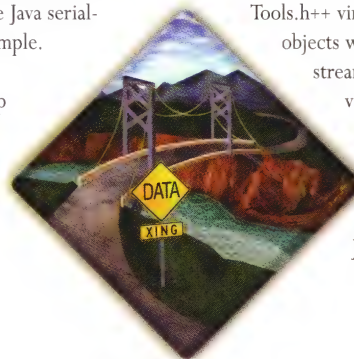
- Includes updated data structure and algorithm classes, plus string, numeric limits, complex classes, and allocators
- Chosen by many major vendors, including Borland, Sun, DEC, SGI, and HP, for inclusion with their standard-compliant C++ compilers

C++ and Java™



We make it easier for you

Serialize.h++ gets your C++ and Java apps to “speak the same language.” Make changes only to your C++ apps—there’s no need to change your Java code. When you use our C++ implementation of the Java serialization mechanism, your C++ apps can exchange objects and data with any Java app that uses the Java serialization mechanism. It’s that simple. Plus, we include pre-built ReaderWriter classes that map popular Tools.h++ classes to corresponding classes in the Java core libraries.



JTools 2.0 includes Java virtual streams, a set of classes for exchanging objects between your Java and C++ programs with a minimal amount of programming. Virtual streams work on both sides of the equation: Java objects are serialized into Java virtual streams using Tools.h++ virtual streams format, and the objects written with Tools.h++ virtual streams can be decoded using Java virtual streams. JTools 2.0 includes a set of pre-built Streamers that map Tools.h++ classes to Java core library or JTools classes.

apps not on speaking terms?

Rogue Wave gets them talking.

C++ and Java are different languages... so how do you exchange data between them? With Rogue Wave's C++/Java Interoperability Suite, you can get your apps to talk to each other, so you can take advantage of the portability of Java while continuing to leverage the benefits of C++ in your applications.



Both JTools 2.0 and Serialize.h++ give you the power and flexibility to get your Java and C++ apps communicating. When you use them along with the other products in the Interoperability Suite, you get reliability and portability, plus you save time and effort by using (and reusing) our well-designed classes. Rogue Wave also offers proven, world-class products for developing visual, math, database, and distributed apps, as well as support and training to help you get the most from your development efforts.

When you use **Tools.h++** to build your C++ apps, you get over 120 tested and proven foundation classes including sets, bags, sorted collections, strings, dates and times, and virtual streams. Even if you don't face interoperability challenges now, you can use the virtual streams in Tools.h++ to build C++ apps

that will be ready to communicate with Java apps that use the virtual streams in JTools. Tools.h++ also provides a higher-level OO interface to the Standard C++ Library.

When you add Rogue Wave's implementation of the **Standard C++ Library** to your development toolkit, you not only get a

head start on ANSI compliance, but you also unleash even more useful features in Tools.h++ (such as extended regular expressions). Plus, Standard C++ Library classes are multithread-safe so your code can run in multithreaded environments.

Find out how Rogue Wave's Interoperability Suite can help you realize the "write once, run everywhere" promise of Java while continuing to leverage the benefits of C++ in your apps.

For more information, call us at (800) 487-3217 or visit our Web site at www.roguewave.com

Telephone us in Europe: ROGUE WAVE SOFTWARE GmbH: +49(6103)59 34-0 ♦ ROGUE WAVE SOFTWARE B.V.: +31-20-695 55 11
♦ ROGUE WAVE SOFTWARE S.A.R.L.: +33-1-47695803 ♦ ROGUE WAVE SOFTWARE-UK. LTD.: +44 118 9880224

CIRCLE NO. 215 ON READER SERVICE CARD

Rogue Wave products are now available for purchasing and downloading at our web site.

A-DDJ-797

The Software Parts Company™

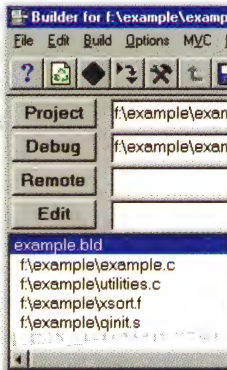


Rogue Wave
SOFTWARE

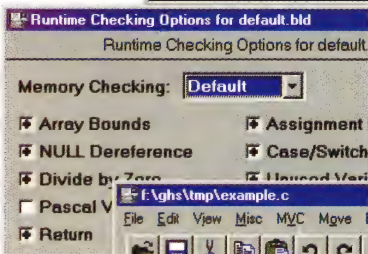
MULTI[®]

The Most Advanced Software Development Environment for Embedded Applications

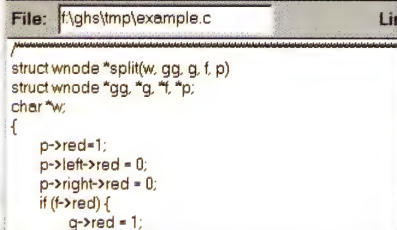
Program Builder



Run-Time Error Checking

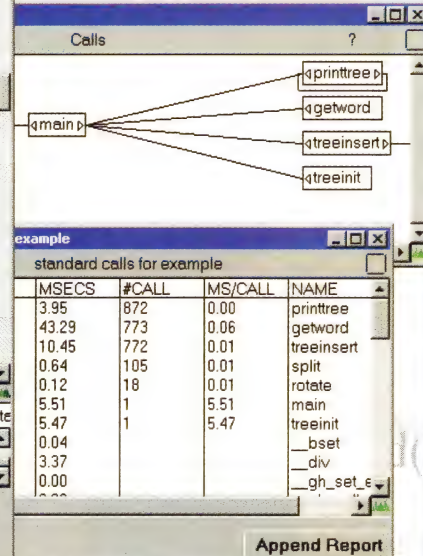


Editor



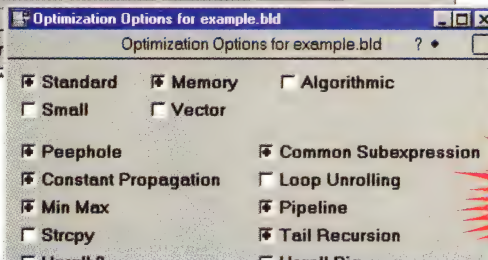
Source Debugger

Class and Program Browser



Execution Profiling

Project Control



**Now Available for
BDM & JTAG
On-Chip Debugging**



• SOFTWARE, INC. •

Software Development Tools for Embedded Development

HIGH PERFORMANCE OPTIMIZING COMPILERS

C C++ Fortran Pascal Ada

THE MULTI[®] DEVELOPMENT ENVIRONMENT

Source Debugger Profiler Run-Time Error Checking
Version Control Editor Source and Class Browser
Program Builder Project Control

CROSS DEVELOPMENT TOOL CHAIN

Assembler Librarian Run-Time Libraries
Linker Utilities Object Code Converters

HOST PLATFORMS

PC/Windows Sun/SunOS HP/HPUX DEC/Alpha AXP
PC/NT Sun/Solaris IBM/AIX SGI/Irix

PROCESSOR FAMILIES SUPPORTED

680x0/683x0	PowerPC	Alpha	V800
R3000/R4000	386/486	SPARC	SH
88100/88110	i960	FR20	M16
RAD6000	RH32	ARM	more...

REAL TIME OPERATING SYSTEM SUPPORTED

INTEGRITY velOSity VxWorks Chorus
pSOS+ VMEexec Nucleus

TARGET SYSTEMS

Simulators ROM Monitors CPU Boards Emulators

Green Hills Software, Inc.

30 West Sola Street
Santa Barbara, CA 93101

Email: sales@ghs.com
http://www.ghs.com

Tel. (800) 856-0715
Fax. (805) 965-6343

Copyright 1996 Green Hills Software, Inc. MULTI is a registered trademark of Green Hills Software, Inc. All other trademarks are acknowledged to their respective owners.

CIRCLE NO. 367 ON READER SERVICE CARD

(continued from page 55)

```
2295 REM Read data and compare
2300 port1=255: TEMP=PORT1: IF TEMP <> XBY(4200H+COUNT) THEN GOTO 5000
2305 EX=EX .OR. 1: XBY(EXAD)=EX
2307 EX=EX .AND. 00FEH: XBY(EXAD)=EX
2310 NEXT COUNT
2320 PRINT "Chip Programmed Successfully..."
2330 GOTO 5000
3000 REM Error in HEX file
3010 PRINT "Error in Hex File. Aborting...": END
4000 REM File not suitable for AT89C2051
4010 PRINT "Hex file not suitable for AT89C2051... Aborting": END
5000 print "COUNT=", count : print temp
5005 CALL 4C03H : CALL 4C09H: CALL 4C0FH: CALL 4C15H
5010 PRINT "Turn Power OFF...": PRINT TIME: CLOCK0
5020 END
```

Listing Two

```
CPU 8051
ORG 0000

mov a, #0; clear accumulator
for_ever: mov pl, a; transfer to Port1
          inc a
          ljmp for_ever; loop for ever.

end
```

Listing Three

```
#include <stdio.h>
#include <dos.h>
#include <conio.h>
#include <process.h>
#include <time.h>

void main(void)
{
    int dport_lpt1, cport_lpt1, sport_lpt1, del;
    unsigned char adc_status, adc_val, cport, chan_num;
    unsigned char nib_0, nib_1, nib_2, nib_3, temp, nibble[5];

    /* Sign ON */
    clrscr();
    printf("MAX186 ADC interface for printer adapter using AT89C2051");
    printf("\nD.V.GADRE");

    /*Get LPT1 port addresses */
    dport_lpt1 = peek(0x40, 0x08);
    if (dport_lpt1 == 0)
    {
        printf("\n\nLPT! not available... aborting\n\n");
        exit(1);
    }
    printf("\nLPT1 address = /X", dport_lpt1);
    cport_lpt1 = dport_lpt1 + 2; /* control port address */
    sport_lpt1 = dport_lpt1 + 1; /* status port address */

    /*initialize the ADC strobe signals*/
    cport=04;
    outportb(cport_lpt1, cport);

    /*clear the status of controller*/
    /*this generates a low going pulse on C1* pin of the control port*/
    cport=cport | 02;
    outportb(cport_lpt1, cport);
    cport=cport & 0xfd;
    outportb(cport_lpt1, cport);

    /* check if ADC is connected & working*/
    adc_status=inportb(sport_lpt1);
    adc_status=adc_status & 0x08;

    if(adc_status == 0) printf("\nADC connected\n");
    else {printf("\nADC not connected...Aborting"); exit(1);}

    chan_num=0; /*set channel number*/
    for(;;)
    {
        /*set channel number to the controller*/
        outportb(dport_lpt1, chan_num);

        /*trigger the controller to read the channel number*/
        cport=cport | 01;
        outportb(cport_lpt1, cport);

        /*wait till it reads the channel number and completes conversion*/
        adc_status=inportb(sport_lpt1);
        adc_status=adc_status & 0x08;
        while(adc_status == 0)
        {
            adc_status=inportb(sport_lpt1);
            adc_status=adc_status & 0x08;
        }
        /*remove the trigger pulse. take it igh to remove the trigger*/
        cport=cport & 0xfe;
        outportb(cport_lpt1, cport);

        /*clear the status of controller*/
        cport=cport | 02;

        outportb(cport_lpt1, cport);
```

```
cport=cport & 0xfd;
outportb(cport_lpt1, cport);

/*now read the converted data*/
for(temp=8; temp<11; temp++)
{
    /*set data port to read nibble 0, 1 and 2 in that order*/
    outportb(dport_lpt1, temp);

    /*trigger the converter to perform the read nibble process*/
    cport=cport | 01;
    outportb(cport_lpt1, cport);

    /*wait till it completes the task*/
    adc_status=inportb(sport_lpt1);
    adc_status=adc_status & 0x08;
    while(adc_status == 0)
    {
        adc_status=inportb(sport_lpt1);
        adc_status=adc_status & 0x08;
    }

    /*remove the trigger pulse*/
    cport=cport & 0xfe;
    outportb(cport_lpt1, cport);

    /*clear the status of controller*/
    cport=cport | 02;
    outportb(cport_lpt1, cport);
    cport=cport & 0xfd;
    outportb(cport_lpt1, cport);

    /*read the nibble and store it temporarily*/
    adc_val=inportb(sport_lpt1);
    nibble[temp-8]=(adc_val ^ 0x80) >> 4;
}

/*construct the full 12 bit number from the stored nibbles*/
printf("\n /d mV", nibble[0] + 16*nibble[1] + 256*nibble[2]);
sleep(1); /*sleep for 1 sec*/
}
```

DDJ

Real-Time Multitasking Tools for Embedded Systems and DOS

RTKernel

Professional, high-performance real-time multitasking system for DOS and 16-bit Embedded Systems.

For Borland C/C++, Microsoft C/C++, and Borland Pascal.
Libraries: \$550 Source Code: add \$500

RTTarget-32

Cross Development System for 32-bit Embedded Systems.

Supports Intel 386 and higher, as little as 16k RAM/ROM.
For Borland C/C++, Microsoft C/C++, and Watcom C/C++.

Libraries: \$1700 Source Code: add \$1000

RTKernel-32

Professional, high-performance real-time multitasking system for 32-bit Embedded Systems.

Supports Intel 386 and higher.

For Borland C/C++, Microsoft C/C++, and Watcom C/C++.

Libraries: \$1950 Source Code: add \$1650

In North America, please contact:

On Time
88 Christian Avenue, Setauket, NY 11733, USA
Phone (516) 689-6654
Telefax (516) 689-1172
email info@on-time.com

NO ROYALTIES! FREE DEMO DISK!
<http://www.on-time.com>

Other Countries:

On Time Informatik GmbH
Hofweg 49, 22085 Hamburg, GERMANY
Phone +49-40-2279405
Fax +49-40-2279263
email to2212.3101@compuserve.com

On Time

REAL-TIME AND SYSTEM SOFTWARE

Thread Pools and Server Performance

Thread pools can save CPU cycles

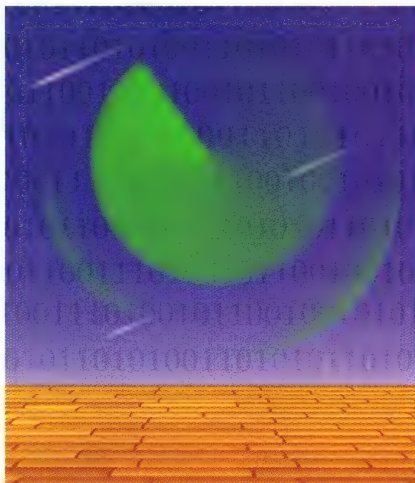
John Calcote

Network service protocols such as HTTP, NNTP, and SMTP have traditionally been implemented on UNIX platforms as heavyweight process servers. When such a server receives a request, it starts a child process, passing it the request and the client connection handle via some form of interprocess communication—usually command-line arguments, environment variables, and inherited file handles. This child process services the request, then terminates. These UNIX environments are optimized for this sort of activity. Processes are cached in memory and code segments are reused for much-improved efficiency. Still, it is fairly time consuming in CPU terms to create the process-control block, and maintain the run-time resources and bookkeeping information for each executing process.

Today's operating systems emphasize threads over processes. Under Windows NT, for instance, creating a new process is much more CPU intensive than creating a new thread of execution in an existing process. Furthermore, operating systems now give you more control over the scheduling of threads than of processes. Even so, starting a new thread can carry a fair amount of overhead with it as well.

John, an engineer on Novell's Directory Services team, can be contacted at JCALCOTE@novell.com.

Some HTTP servers on the World Wide Web service hundreds of thousands of requests per day. If such a server processes only 100,000 requests in a 24-hour period, each request must be serviced in an average of 0.864 seconds. Such servers cannot afford to use even a quarter second creating a new thread for each incoming client request. In reality, many of these requests are being serviced simultaneously. Even UNIX process-oriented servers run several copies of the child process request handler at a time.



In actuality, the number of requests per second varies from minute to minute. Sometimes, requests come in so fast the server doesn't have time to accept the connection in a reasonable amount of time. Clients experience this in the form of a timeout, usually accompanied by a message to the effect that the server is currently unavailable. This all depends, of course, on the ability of the client to correctly interpret a lack of response from a server. Some client applications simply tell

users that the server cannot be found, which can be misleading, if not annoying.

A well-written server application is optimized in two important areas. First, it can process connection requests as fast as possible. Note that a connection request refers specifically to the connection indication event generated by the underlying transport service provider. Second, it can be tailored by the administrator to politely reject all simultaneous connection requests above a certain threshold number.

Tried and True

Typically, a server application is written with a listen thread for each port or socket to be monitored. This thread might sit in a loop sleeping on a call to an operating-system socket monitor function, such as the Winsock *select* or TLI *listen* function. When a connect indication is received, this function wakes up and returns control to the monitor thread, which then attempts to receive the indication (or data, in the case of a datagram protocol). Once the data or indication is retrieved, it must be processed, but the monitor thread needs to do this as quickly as possible to get back to listening for more connection indications. Remember, if the monitor thread is not blocking on a call to *select*, then it's not listening for connection requests. Example 1 is pseudocode that depicts a commonly used monitor algorithm.

What the server thread does with the client's connection handle is of no concern to the monitor thread. As far as the monitor is concerned, the client has been serviced. Although this looks fairly efficient at first glance, the acts of retrieving the connect indication, establishing the connection, and starting the new server thread can be overly time consuming for a truly high-performance server. It would



The first RAD C++ tool that's all pluses. No minuses.

- + Component-based RAD C++
- + Drag-and-drop programming
- + Easily create reusable components
- + Distributed application development
- + NetImpact Dynamo for creating data-driven Web applications
- + Fast database access with the Powersoft DataWindow™ and native drivers
- + Sybase SQL Anywhere™ server
- + Support for legacy MFC code
- + ObjectCycle™ for team development



Productivity without compromise. New Power++™ 2.0.

Introducing Power++ 2.0. The powerful new version of Optima++.™ It's the RAD C++ tool that delivers the performance you need for client/server, distributed and Internet applications—plus the productivity you need to get the job done.

Power++ is an ideal environment for building the server-side of Internet projects. It helps you minimize the development time of rich, dynamic, data-driven Web applications using Sybase® NetImpact™ Dynamo.

And with its innovative component-based approach, Power++ makes it easy to build scalable applications. You can quickly build high-performance ActiveX™ Server components and deploy them in distributed frameworks such as Powersoft Jaguar CTS™ and Microsoft® Transaction Server™.

What's more, with Power++ you can minimize development time while maximizing your existing investments. You can reuse over 245 components and

"Power++ is easily the most productive C++ environment we've used. In only seven months, we've rolled out diverse business applications from mission-critical manufacturing solutions to sales force automation."

—Paul Ackley, Programmer/Analyst, United McGill Corporation

classes—or easily create your own. You can even leverage your legacy MFC and ANSI C and C++ code.

Award-winning Power++ combines all of these proven features and more. Check it out today.

 **Powersoft.**
Open Tools from Sybase, Inc.



Download your free Test Drive Edition today: www.powersoft.com or 1-800-395-3525.

©1997 Sybase, Inc. All trademarks are property of their respective holders. All rights reserved.

CIRCLE NO. 295 ON READER SERVICE CARD


```
while server is up {
  listen for connect indication
  retrieve connection indication
  connect to client
  start server thread -- pass client connection
}
```

Example 1: Monitor algorithm pseudocode.

```
listen for connect indication
tell someone else to start listening again
retrieve connection indication
connect to client
service client request
terminate
```

Example 2: Thread-pool-based monitor pseudocode.

(continued from page 60)

be nice if one could write code such as Example 2, a thread-pool-based monitor. Essentially, the server application maintains a pool of threads that may be used to execute the monitor function. Each time the currently executing monitor thread returns from a call to select, the first thing it does is reschedule itself with the thread pool, starting a new copy of the monitor thread and blocking on select. Meanwhile, the original monitor thread continues, servicing the client request to completion.

Listing One (listings begin on page 86) is a simple thread-pool-based server program that monitors a well-known socket for client requests. This server uses a

connectionless protocol, which, in this case, means that the client passes all request data in a single initial packet (called a "datagram"). The server responds by also returning a single packet. In this protocol, the client must send the five-character string "Ping" (including the terminating NULL character). If the client sends the correct request, the server answers by returning another five-character string ("Pong"); otherwise, the error message "101 Invalid request" is returned. Granted, this is a bit contrived, but the general process is the basis for nearly all standard high-level, text-based Internet protocols. (Ping.h and related files are available electronically; see "Availability," page 3.)

Want to build applications for the Net?

Distinct provides the most comprehensive, robust and market tested Internet and Intranet components available in the world. Just plug them into your applications and deliver solid products fast and on schedule. Every time.

Share in the experience:

"We conducted a test session to compare the performance of similar products, and Distinct's product was better. Distinct's Internet Development Kit allowed us to have the communications layer of the package ready in record time. We gained much more flexibility than expected due to the number and quality of the objects contained in the package."

-Dr. Shyam Sunder, Carnegie Mellon University

"Distinct provided a Telnet OCK/VBX that saved up to 6 months of development time and reduced the overall development cost."

-Paul Calboun, Tandem Computers

"The Distinct package includes custom controls that are easy to use, reliable, and perform well."

-Darwin Hatbaway, 3M Company

"By using Distinct, CRM saved a lot of time and money and provided great solutions for challenging tasks."

-William Gutekunst, CRM Technologies

"It is not often, in today's market, that you can find companies that want to find the solution to a customer's problem, no questions asked. Thank you."

-Scott G. Phillips, NTN Communications, Inc.

Distinct Visual Internet Toolkits: just what you should expect from the most respected and experienced net components company in the business.

distinct
The world leader in Internet development tools.

http://www.distinct.com
sales@distinct.com
Phone: 1-408-366-8933
Fax: 1-408-366-0153



Pools of Threads

I've based my thread pool's interface on Novell's Asynchronous Event Services (AES) API. The AES API essentially contains the two functions in Example 3(a), with AESEVENT being defined as Example 3(b).

In Novell's implementation, the AES-EVENT structure passed to *ScheduleAES-Event* must be declared in a manner that assures its validity for the life of the event. This is because the data structure instance itself is linked (by way of the *next* field) into the system's internal list of events waiting to be serviced, then passed to the event handler as event data. This may seem a bit odd, but the reasons for this sort of one-track, efficiency-minded attitude at Novell are historical in nature—and not necessarily bad—in most instances. There are significant advantages to passing the event's AESEVENT data structure to the event handler. In the first place, nothing can go wrong with *ScheduleAESEvent*—thus, the void return value. It doesn't even allocate memory. To pass private event data, you can allocate your structure instance larger and define your own data fields after the system-defined fields.

Cancelling an event is simply a matter of passing the address of the same structure instance to *CancelAESEvent*. If the event has not already been scheduled, and thus removed, from the waiting event list, it can be found and removed using the pointer address as a lookup field. If it has already been scheduled, then the allocated thread is already running your handler, and must be terminated in a more proprietary manner. This is indicated by the return value of *CancelAESEvent*.

My version of this interface encapsulates all of the necessary data for an entire thread pool in a C++ class called *AES* (see Listing Two). This class contains two private internal class definitions, one

Licensing fees required for redistribution. Some protocols may not be available in all forms. Distinct is registered trademarks of Distinct Corporation. Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation. Copyright Distinct Corporation, 1997.

CIRCLE NO. 125 ON READER SERVICE CARD


```
(a) void ScheduleAEEEvent(struct AEEVENT *aes);
    int CancelAEEEvent(struct AEEVENT *aes);

(b) struct AEEVENT {
    struct AEEVENT *next;
    long wakeupTime;
    void (*func)(struct AEEVENT *);
    ...
};
```

Example 3: AES API functions.

```
void Advertise(void *ref) {
    SAP& This = *(AES*)ref;
    This.BuildPacket();
    This.SendPacket();
    This.aes.ScheduleEvent(60, Advertise, ref);
}
```

Example 4: The broadcast function.

called *PoolThread* and the other, *Event*. *PoolThread* is used to manage a single pool thread, while *Event* is used to store event information for each scheduled event. Events are scheduled by the *ScheduleEvent* member function, which takes a delay time in seconds, a pointer to an event-handler function, and a generic pointer to data to be passed through to the event handler. It returns a handle to an event, which is implemented internally as a cast from the allocated data-block pointer. This handle may be passed to the *CancelEvent* member function. My version of *ScheduleEvent* can fail in low-memory conditions. One of the main reasons I chose to implement this API as a C++ class is the freedom it allows users to reimplement internal functionality without noticeably affecting the interface.

Listing Two shows my implementation of the AES class and, subsequently, its private classes. The pool monitor and pool thread-control functions are static members of the *PoolThread* and *AES* classes (respectively, *Monitor* and *Worker*). Because they are static members, they have no implied *this* pointer. I solve this problem by passing the instance pointer as the thread data for each thread created. The pool monitor gets a pointer to the *AES* instance, while the pool thread gets a pointer to its *PoolThread* instance.

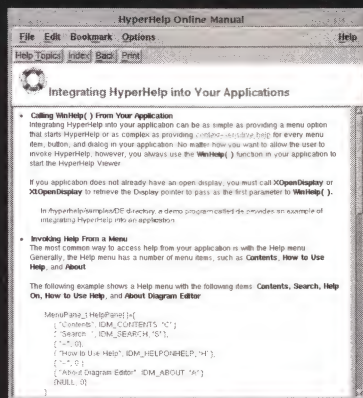
The AES constructor takes a single parameter—the number of threads to create on instantiation. This parameter defaults to 8 if left unspecified. When an event is scheduled, a new event block is allocated, initialized, and inserted into the AES waiting list. Then the monitor

semaphore is released and an event handle is returned to the client. Releasing the monitor semaphore wakes up the monitor thread, which rescans the list of waiting events for the nearest event due. If there are no events ready to be scheduled, the monitor sleeps on its semaphore for a time period calculated to wake it up when the closest event is due. If events are due, but no pool threads are available, a debug message is posted, indicating that the pool needs to be tuned, and the monitor sleeps until a new thread becomes available. Simply sleeping may seem inefficient for a low-thread scenario, but when tuned properly, the pool will rarely run out of threads during a session. If it runs out too often, you need to increase the number of threads in the pool for your application.

ONLINE HELP

For UNIX & OpenVMS

HyperHelp™ is the UNIX and OpenVMS equivalent of the industry-standard Microsoft Windows WinHelp. Use your favorite Windows help authoring tool to create online hyperlinks, pop-up definitions, graphics, full-text searching, powerful macros, and a wealth of other features. You can even utilize the same RTE, HPJ and graphic files used to create your Windows help.



HyperHelp also supports Frame-Maker MIF, HTML, SGML and ASCII text as authoring file formats for online help. Best of all, the HyperHelp Viewer can be distributed royalty-free to your customers. For more information, call (203) 438-6969, fax (203) 438-5013 or download a free demo from ftp.bristol.com. In Europe call: +31 (0)33 450 50 50 or fax +31 (0)33 450 50 51.

Now
Supports
HTML!

All trademarks are the property
of their respective holders.

Web: www.bristol.com
Email: info@bristol.com

BRISTOL
TECHNOLOGY

CIRCLE NO. 170 ON READER SERVICE CARD

YOU CAN'T GET MORE GRAPHIC THAN THIS.

Introducing the Essential Books on Graphics Programming CD-ROM from the Dr. Dobb's CD-ROM Library

Need to get up to speed on graphics programming—fast? There's only one definitive source you can turn to—Dr. Dobb's *Essential Books on Graphics Programming CD-ROM*!

Get seven of the most essential books in graphics programming, with full text, diagrams, graphics, and source code—all on one CD-ROM. From fundamental algorithms to the most complex techniques, this CD-ROM lets you find all the critical information you need for your graphics programming projects.

Texture mapping, color modeling, and morphing, all optimized for speedy 2-D and 3-D graphics programming. Practical image acquisition and processing. High-quality 3-D photorealistic graphics. Digital halftoning. Image synthesis and special effects.

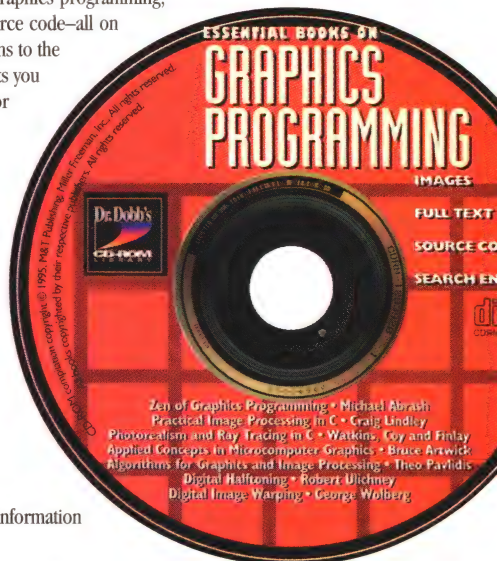
And much more!

Plus, all the source code on the disks supplied with the books are included.

Your CD-ROM comes with must-have features:

- A powerful search engine for locating information across all or selected books
- Tables of contents and indexes hotlinked to their topics
- Copy, bookmark, and notes functions

And you get a full money-back guarantee!



Windows Platform

The DDJ Selection: We did the legwork for you!

<i>Zen of Graphics Programming</i>	Michael Abrash
<i>Photorealism and Ray Tracing in C</i>	Watkins, Coy and Finlay
<i>Practical Image Processing in C</i>	Craig A. Lindley
<i>Applied Concepts in Microcomputer Graphics</i>	Bruce Artwick
<i>Digital Halftoning</i>	Robert Ulichney
<i>Digital Image Warping</i>	George Wolberg
<i>Algorithms for Graphics and Image Processing</i>	Theo Pavlidis

Order Your CD-ROM Today!

ONLY \$69.95
PLUS SHIPPING



FAX ORDERS: 913-841-2624
E-MAIL ORDERS: orders@mfi.com
MAIL ORDERS: Dr. Dobb's CD-ROM Library,
1601 West 23rd St., Ste. 200
Lawrence, KS 66046-2703 USA
INTERNATIONAL: USE MAIL, FAX, E-MAIL,
OR CALL 913-841-1631

More than Meets the Eye

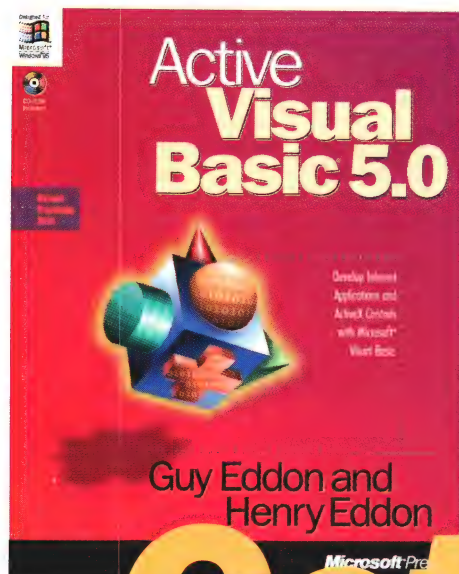
The uses of this AES-based implementation of a thread pool extend beyond those I've established here. This thread pool may also be used to schedule timed events. For example, Novell's Service Advertising Protocol (SAP) uses AES to broadcast a service-advertising packet every 60 seconds. Using class AES, the broadcast function can be written like Example 4. In this example, presumably there is a *SAP* class instance, which contains an *AES* instance member and whose pointer has been passed as event data to this event handler. The *AES* object might just as well have been declared as a static global, accessible to this handler by virtue of its definition in this module.

There is a lot of room for improvement in this thread pool. The stack size might be an issue in your application, and thus become a tunable parameter to the *AES* class or the individual threads. The one problem this might present would show up during creation of the thread pool in the *AES* constructor. The pool is allocated as an array of *PoolThread* objects. If you need to pass a stack-size parameter to the constructor for the *PoolThread* objects, you will not be able to use array allocation. One solution to this limitation is to allocate raw memory for the thread-pool array using operator *new*, then use *placement new* to instantiate a *PoolThread* object for each block in the array. If you do this, you should call the destructors for each *PoolThread* directly, then deallocate the raw memory using operator *delete*. Another possible feature that the pool could provide is thread-local storage on platforms that support it for maintaining thread-specific data.

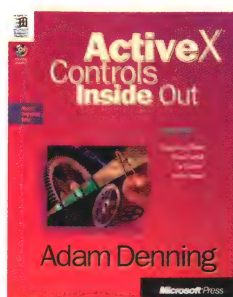
However you choose to implement a thread pool, it's important to keep a few things in mind. First, this pool's monitor thread does not poll the waiting list during idle time. Neither do the pool threads poll their respective event pointers. When not working, they are always sleeping on a semaphore. Virtually no CPU overhead is unnecessarily maintained. The whole idea is to improve efficiency. In the high-performance server discussed earlier, it was critically important that the least amount of cycles be used between the current thread's return from *select* and the next thread's call to *select*. Wasting cycles during a call to *ScheduleEvent* defeats the purpose of the thread pool. For this reason alone, you might consider reimplementing the thread pool using Novell's strategy of linking the event blocks right into the system's waiting list.

DDJ
(Listings begin on page 86.)

Dr. Dobb's Journal, July 1997

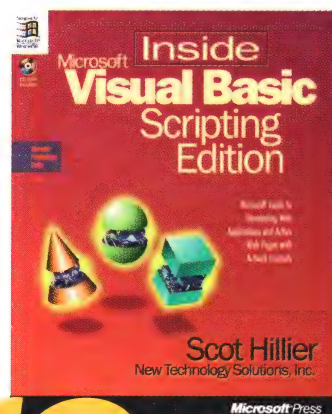


Active Visual Basic® 5.0
\$39.99 (\$54.99 Canada)
1-57231-512-1



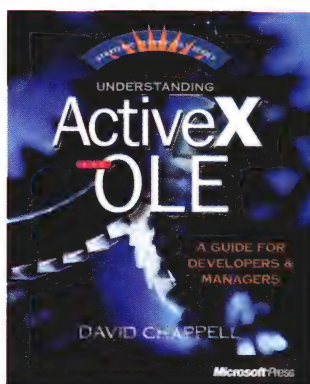
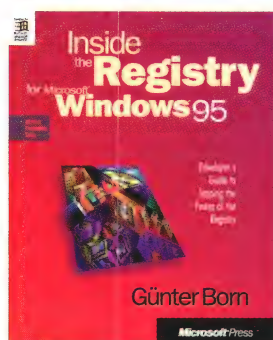
ActiveX™ Controls Inside Out,
Second Edition
\$39.95 (\$54.95 Canada)
1-57231-350-1

Inside Microsoft® Visual Basic®,
Scripting Edition
\$39.95 (\$54.95 Canada)
1-57231-444-3

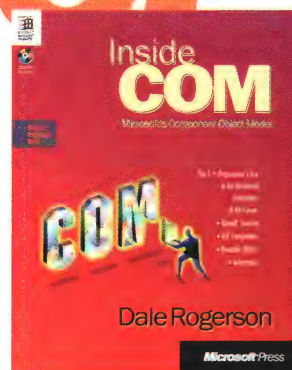


Get there from here.

Inside the Registry for
Microsoft® Windows® 95
\$24.99 (\$34.99 Canada)
1-57231-424-9



Understanding ActiveX™
and OLE
\$22.95 (\$30.95 Canada)
1-57231-216-5



Inside COM
\$34.99 (\$46.99 Canada)
1-57231-349-8

Let these **authoritative** Microsoft Press® books show you the way when you develop for Windows 95 or the Internet: **INSIDE THE REGISTRY FOR MICROSOFT WINDOWS 95**—hard-to-find information that helps you **create** well-behaved applications and give greater functionality to Windows 95; **INSIDE COM**—the foundation of both OLE and ActiveX, vital **technologies** for future application development; **ACTIVE X CONTROLS INSIDE OUT**—an in-depth guide to building powerful custom **controls** and componentware; **INSIDE MICROSOFT VISUAL BASIC, SCRIPTING EDITION**—how to leverage your Visual Basic knowledge into **Internet** results immediately; **ACTIVE VISUAL BASIC 5.0**—new ways of using Microsoft Visual Basic version 5.0 to create Internet-enabled applications and interactive Web content; and **UNDERSTANDING ACTIVE X AND OLE**—the strategic significance of the Component Object Model (COM) as the foundation for Microsoft's **object technology**. You're on your way.

Microsoft® Press

Available in quality bookstores and computer stores worldwide.

To locate your nearest source for Microsoft Press® products, reach us at
1-800-MSPRESS in the U.S., or at **mspress.microsoft.com**





Sounding Off with the RSX Library

3-D audio for high-performance applications

Steve Durham

If you close your eyes and have a friend walk around you snapping their fingers behind you, above you, or below you, you can pinpoint where the sound is coming from, even though your eyes are closed. Working together with your ears, your brain is able to determine where the sound is coming from because sounds phase shift and attenuate differently, depending on the location of the source.

Three-dimensional sound attempts to synthesize this effect in software or hardware. This is analogous to a graphical rendering performed by 3-D graphics libraries.

Intel's Realistic Sound Experience (RSX) library was initially developed to enhance the silent 3-D world of VRML. (The VRML 2.0-compliant RSX 2.0 SDK, which can be downloaded at <http://developer.intel.com/ial/rsx/>, requires Windows 95/NT, a Pentium or 33/486DX, and stereo PC soundcard. MMX is not required, but it significantly enhances performance.) As the RSX project developed, however, it was expanded to support more demanding applications of 3-D audio, such as 3-D games. These capabilities include HRTF-based (Head Related Transfer Function) localization, Doppler, reverberation, and real-time streaming.

RSX uses a set of high-level COM-based interfaces for describing a 3-D audio world that follows a model similar to that used by high-level graphical rendering engines. A typical 3-D graphical application describes a scene with objects and defines a view of the scene by specifying a camera. When components in the scene change, the application informs the rendering en-

gine, and the application renders a new image from the defined camera's perspective.

Similarly, an application describes an audio scene with sound-source objects called "emitters" and defines an audio "point-of-view" by specifying a "listener." When emitters change position or orientation, or when the listener moves, the application renders a new audio "view" from the listener's perspective.

Like graphical objects, you place and orient sounds in 3-D space. You can adjust other properties, such as the definition of ambient and directional sound regions to achieve the desired rendering effect. When you finish defining listeners and emitters, the RSX library can begin an audio rendering of the scene. For best results, keep the audio environment consistent with the graphical environment. When objects move, both the graphical and audio rendering engines need updating. Likewise, when the view of the scene changes, the camera and listener positions need updating.

Components of 3-D Sound

Several components contribute to the rendering of 3-D sounds. "Spatialization" is the process of placing a sound in 3-D space. This is the audio counterpart to the position of an object, but rather than rendering with perspective and shading cues, audio is rendered such that sounds appear to originate from a specific location in space. RSX implements several localization algorithms, the simple one being a variation of left/right panning. The panning algorithm provides left and right placement of sound, but lacks what most would consider 3-D localization (the ability to simulate left/right, up/down, and

front/back cues). To achieve true 3-D localization, RSX uses an HRTF-based algorithm.

"Reverberation" is an effect for simulating acoustic environments, ranging from small chambers to open canyons. This is the audio equivalent of environmental effects such as fog. Like fog rendering, reverberation has a global effect on the entire scene. The RSX reverberator is based on the Schroeder reverb algorithm, which can simulate a wide range of acoustical environments. Some parameters are user adjustable, such as intensity and delay. A minimal level of reverberation greatly improves the perceived spatialization.

"Doppler" is used to simulate sounds in relative motion. The graphical counterpart to Doppler is motion blur. Rather than blurring an image, an audio signal is "blurred" through pitch shifting. As the velocity of the moving object dictates the degree of motion blur, so does it dictate the degree of pitch shifting. The classic experiment with Doppler involves a horn sounding on a moving train. As the train travels toward you, the sound waves are compressed, effectively increasing the pitch. As the train travels away from you, the sound waves are rarefied, correspondingly decreasing the pitch.

"Pitch shifting" is commonly used to model engines, where the engine's pitch is proportional to the engine RPM. Explicit pitch adjustment is the equivalent of tweaking the color of light sources in the graphical world. The Doppler effect can be viewed as an automatic pitch shifter which translates the relative velocity between an emitter and the listener. RSX implements two pitch-shifting algorithms, a low-quality/low-cost

Steve, aka Bull, a senior design engineer for CodeSign, can be reached at sdurham@cdcorp.com.

Let's get right to the point.

Until now, profilers for 32-bit Windows apps haven't been worth using. They took way too long, delivered very little data, and the info they did provide was hard to read and understand. But now all that's changed.

Introducing HiProf, the first hierarchical profiler for Windows C/C++ Applications

With HiProf you can:

- Clearly see time spent in **parent functions and their children** as well as all the calls taking place between them
- Profile your binary file in **just minutes** (no OBJ files or source code required)
- Selectively exclude dlls or portions of your exe from profiling for added efficiency and to **evaluate third-party components**
- View profiles in **intuitive graphical displays** that allow you to drill down easily into function relationships and even into the source code

HiProf identifies all performance bottlenecks so quickly and easily that you'll use it frequently throughout the development cycle. So why not get started with HiProf right now?

Download the **FREE HiProf trial version** from www.tracepoint.com or call 888-688-2504 for more information.

TracePoint

HiProf

Free Trial
Version

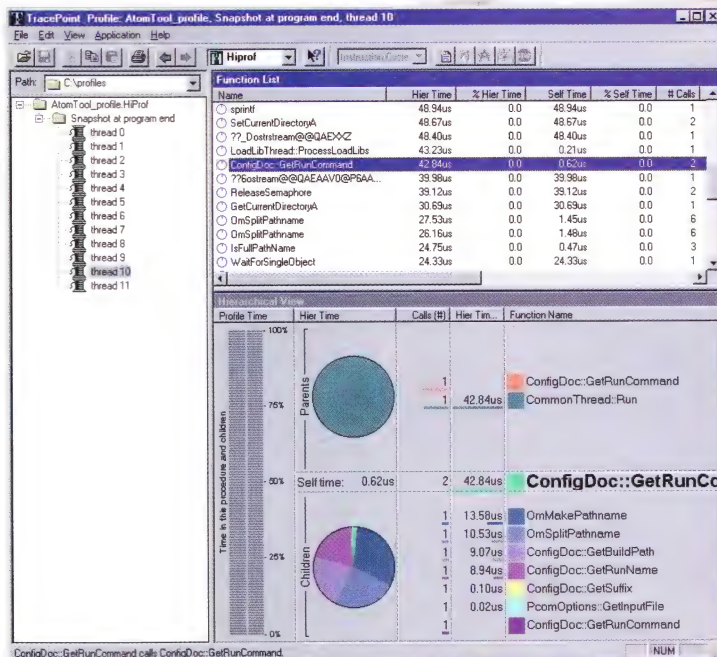
@

www.tracepoint.com



TracePoint: A DIGITAL Company

digital



Copyright © 1997 TracePoint Technology Inc. All rights reserved. TracePoint and HiProf are trademarks of TracePoint Technology Inc. All other products are trademarks of their respective owners.

CIRCLE NO. 359 ON READER SERVICE CARD

(continued from page 66)

drop-and-dupe algorithm and a high-quality/higher-cost pitch shifter without the artifacts of the low-cost implementation.

Sound Emitter Model

RSX models an emitter in terms of two concentric elliptical regions. These two ellipses describe an ambient region and an attenuation region. The shape of these ellipses can be adjusted to vary the directionality of the emitter. Figure 1 illustrates the elliptical sound model.

The inner ellipse, identified by a minimum front range and minimum back range, defines an ambient region. The ambient region maintains a constant, maximum intensity relative to the given sound. The outer ellipse, identified by the maximum front range and maximum back range, defines the attenuation region. Within this region, audio is localized, and the intensity decreases logarithmically. Beyond the outer ellipse, the intensity is zero.

Audio Streaming

RSX provides interfaces to define two types of listeners and two types of emitters: direct and streaming listeners, and cached and streaming emitters.

A direct listener lets RSX output sound to a standard audio interface. A streaming listener provides access to processed buffers from RSX. Both allow you to define a listener and control its position and orientation in the audio environment.

Cached emitters are file-based sound sources. Streaming emitters require buffers and let your application process real-time data. When you create an emitter, you assign it a specific sound, much as you would assign lighting and texture properties to a polygon. You bring sounds to life by controlling when they start and stop and by adjusting their volume and tone.

Coordinate Systems and Units

RSX supports both right- and left-handed coordinate systems. The application simply informs RSX which format to use. The left-handed coordinate system can be presented in the following orientation with respect to the display: the x-y plane is the screen, with x increasing to the right, y increasing upward, and z increasing into the screen. The right-handed coordinate system uses this same x-y orientation, but specifies z increasing away from the screen.

RSX defines all of its audio using an absolute coordinate system, known as "world" or "scene" coordinates to many graphics libraries. There is no support for specifying the position or orientation of one object relative to another object. When working with high-level graphics libraries, coordinates passed to RSX should be in either world or scene coordinates. That is, you must specify an audio object's x-, y-, and z-coordinates.

RSX calculates the relative relationship between objects, their position and attributes, and uses this information for rendering. In this model, distance has no units.

RSX uses the speed of sound for Doppler effects, and all time is specified in seconds. You must supply the value for the Doppler effect in generic units per second.

COM Interfaces

The RSX library consists of five COM interfaces and two abstract base classes. The abstract base classes group together the methods that are common to all emitters or all listeners. The emitters or listeners provide the abstract base class functionality through their respective interfaces. Table 1 lists the RSX interfaces.

WaveSpinner Application

While working on the RSX library, I often found myself creating example .WAV files that were spatialized in different ways to demonstrate different effects. I grew tired

of trying to slap together an application just to move the sounds around. I decided to write a tool that would accept a script that described the movement and characteristics of sounds in a 3-D world. The tool, called "WaveSpinner," reads such a script, animates and spatializes all the sounds using RSX, and writes the result to a .WAV file. This tool is very useful for creating static sequences such as background noise, missile fly-bys, and other sounds that need to be spatialized, but will not change. The missile fly-by, for example, can be used many times in the course of a game, but will require minimal CPU time, as it does not need to be spatialized at run time. (The complete source code and related files for WaveSpinner are available electronically; see "Availability," page 3.)

Example Script

Listing One (listings begin on page 89) is a WaveSpinner script that animates some WAV files commonly found in the Windows 95 C:\Windows\Media directory. You already have these input files available if you have a default install of Windows 95.

To simplify parsing of the script file, I adopted a pseudo-Visual Basic scripting syntax. There are four types of objects: listener, environment, output (which are predeclared), and emitter (which you declare in the script). There can be any number of emitters and only one instance of each of the other objects. For each object, you can set certain variables or call a method. Variables are set with an equal sign, and these settings hold throughout the entire processing cycle; if you try to set a variable twice, the last setting is used. Methods, of course, can be called as many times as necessary. The methods closely reflect the actual RSX API, except that they have an additional first parameter that specifies time. This time parameter adds the animation element to the generated .WAV file. For example, if you specify Listing Two, WaveSpinner will smoothly move the sound from the first position to the second in three seconds. The granularity of time is 40 milliseconds, which is about the smallest practical time interval because RSX needs sufficient buffer length to "smooth" one buffer into the next. If the buffer size is too small, the buffers cannot be blended sufficiently, and the result is unwanted pops and clicks in the output.

A script file will usually declare the output settings up front, although they can be anywhere in the file since WaveSpinner collects all settings before generating output. In the example, I create a 30-second .WAV file and write it to c:\noises.wav, with the output spatialized for headphones.

One of the output settings is *Peripheral-Type*. The possible values for this setting

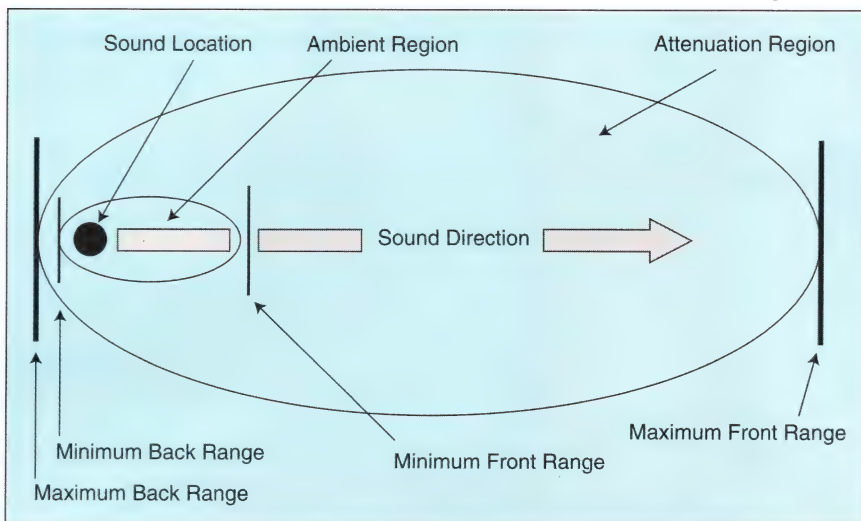


Figure 1: Sound emitter model.

FIND BUGS BETTER

with


Insure++

What is the fastest error detection tool available for detecting bugs? The answer might not be what you expect. Tools that "say", may not always "do". The true answer is really not how *fast* an error detection tool claims it can find a particular bug, but how long it takes to remove the largest number of bugs from your software. The thoroughness of the tool determines, let's say, the length of your next vacation or the length of your employment. The bottom line is, you don't want *any* undetected bugs.

On average, there are 1.2 defects¹ in every 200 lines of code. That is about 12,000 hours -or 5.7 man years!- to debug a program of 200,000 lines of code, costing an estimated half-a-million dollar! The more bugs left undetected by your tool, the more time you will spend finding bugs **manually** (i.e. slowly). You depend on your automatic error detection tools to work the way they are meant to- detecting bugs and detecting them *automatically*.

You expect your tools to work-to improve and to speed up your development process. Unfortunately not all tools live up to their claims; some even slow down your development process! Many tools claim their efficiency and effectiveness. Let me fill you in on a secret - they are not as effective *or* as efficient as they claim. Some of these tools don't go straight to your source code to debug. Instead,

they search and detect the bugs at the object level where much of the critical information about your source code have been lost - irretrievably! As a result, your "fast" debugging tool leaves a Pandora's Box full of bugs that will cause your program to crash, in the worst case, at your customer's site! Not fun, when you realize you have spent so much money (and time) on these tools.

Consider *Insure++* from ParaSoft. It was developed to streamline the process of finding software defects. It is the most accurate and thorough automatic runtime error detection tool available in the software development industry. *Insure++* automatically detects a large variety of errors, including problems such as memory leaks, memory corruption, language specific errors, and much more.

What normally takes 12 hours² to find and fix will only take about 30 minutes with *Insure++*. Its precision and accuracy will catch bugs you don't even know exist and doesn't leave any bugs undetected! This makes *Insure++* a pretty fast detection tool - in fact,

100x's faster than any other tool! With its patented *Source Code*

Instrumentation (SCI) (U.S.

Patent #5,581,696), *Insure++* inserts test and analysis functions around every line of the program's original source code. *Insure++* offers the most optimal checking since it goes straight to the source code. *Insure++* is inherently more accurate at identifying errors than any other tool on the market!

As you can see, SCI is the way to go in detecting bugs—finding them with accuracy and efficiency. Tools that go to your object code will claim speedy detection, but, are limited in detecting many kinds of bugs. With *Insure++*, you are assured that you are maintaining the highest standard of quality software, not to mention extra time to spend writing code, instead of debugging. To learn more about *Insure++*, call toll-free (888) 305-0041. Or download a copy of *Insure++* today from the web: www.parasoft.com/insure to see what error detection is all about.

¹ DeMarco and Lister research
² Microsoft internal study

ParaSoft Corporation

2031 S. Myrtle Avenue, Monrovia, CA 91016 Phone - (818) 305 0041 Fax - (818) 305 9048 Website - www.parasoft.com E-mail - info@parasoft.com

CodeWizard The only software available to assist programmers in achieving effective C++ programming - automatically.

No need for manual labor;
no memorization
of applied rules.

Code Wizard


Insure++

Insure++ is the most thorough error detection tool available for C/C++ programmers. It will find not only your ordinary bugs, but also those hard-to-detect errors that cause major problems. It pinpoint memory leaks, memory corruption, dynamic memory errors, and much more.

Call (888) 305 0041

or Click www.parasoft.com for more information

or to get **CodeWizard** and **Insure++**

(continued from page 68)

are *Headphones*, *Speakers*, and *Transaural*. It is important to know what the output type is because the HRTFs and postprocessing of the .WAV file change depending on the type. If the peripheral type is set to *Speakers*, then the sound is not spatialized using HRTFs, but Doppler, reverb, and attenuation effects still apply. If the output type is *Headphones*, then HRTFs are used. If the peripheral type is *Transaural*, then HRTFs will be applied and the output will be fed through a cross-canceling filter, which is the necessary adjustment to get the 3-D effect from two open speakers.

Listing Three summarizes the scripting API. The README.TXT file included with WaveSpinner describes the methods in greater detail, and shows an example of each.

For the *environment* object, you can set the reverb volume and the speed of sound (which effects Doppler). The listener methods allow you to position and orient the listener. Emitters are declared by specifying the source file; for example, *emitter.MyEmitter1=c:\tidal.wav*. This emitter can then be used in the script. Order of statements in the file is unimportant except in two cases: setting a parameter twice, in which case the second setting will be used; and two statements falling inside a 40-ms time period, in which case both will be called (in order).

Code Overview

WaveSpinner uses three main classes: *CEmitterObject*, *CListenerObject*, and *CEnvironmentObject*. There probably should be an output class as well, and to its nonexistence I can plead no excuse other than laziness and the fact that this code has changed several times before settling into its present form. Anyhow, the existing

HRTF Sound Filtering

A Head-Related Transfer Function (HRTF) is simply a filter you run the sound through, and there is a different filter for every location around your head. In practice, a table with a fixed number of HRTFs is used, and points are interpolated between the existing HRTFs. To measure an HRTF, you typically would place microphones in the ears of human subjects and measure the response to test signals placed on various points of a sphere around the head. A sound coming from behind

the head will have different phase shift, amplitude, and frequency attenuations than a sound coming from a different position, like the front. A set of HRTFs measured from one person may not work very well on another, because their heads and ears are shaped differently, and hence have different transfer functions. Ideally, a 3-D sound library would include several sets of HRTFs, so that the end-user can pick the one that sounds best.

—S.C.D.

classes wrap the actual RSX interfaces, and incorporate additional methods to parse commands from the input script file and to animate the sounds as time marches on.

WaveSpinner makes an initial pass through the script file to gather the output settings and build a linked list of declared emitters. It then runs a second pass for environment, listener, and emitter statements.

To parse the script file, the main parsing loop reads a line from the script file, determines who the line belongs to, and then asks the owner to parse the line themselves. For example, if an incoming line starts with "listener.Position(...)", the listener object will be handed the line using the *CListenerObject::ParseCommand()* method. If the listener has a problem parsing it, it merely returns false, the line is sent as an error to the Wave Spinner main window, and the line is ignored.

Likewise, statements prefixed with "environment." or starting with the name of

an emitter (determined by looking through a *COBList* of emitter objects) are passed to the *ParseCommand()* method of the correct object instance. Internally, the objects examine the information and construct lists (using the *CKey* objects described later) or tables (for calculating splines).

After the file parsing, WaveSpinner adjusts the RSX peripheral type. RSX relies on a registry setting to know what the peripheral type is. To actually adjust the RSX setting, WaveSpinner stores the old setting from the registry, writes the new value, runs the script, and then switches back to the old setting.

WaveSpinner then enters the main play loop. During each iteration of the play loop, WaveSpinner calls the *SetTime()* method of each object. In response to *SetTime()*, each object refers to its internal keys or tables and makes any changes necessary to the underlying RSX object. Once all the objects have been correctly updated, WaveSpinner requests a buffer from the RSX streaming listener. Inside in RSX, the streaming listener gets its input data from the RSX emitters and generates an output buffer. WaveSpinner then takes the buffer and writes it to the output .WAV file.

The play loop then repeats, incrementing the time by 40 ms each iteration. The loop stops when the time exceeds *output.TotalPlayTime*. WaveSpinner then deletes all RSX objects, closes the output file, and restores the peripheral setting in the registry.

Keys versus Splines

The listener and emitter classes both use *CKey* and *CSplineArray* objects. The environment object uses only *CKey* objects. While the script is running, there are two types of settings that will be adjusted while time marches on—those that need to be interpolated and those that don't. The only things that need to be interpolated are position and orientation. Other settings, such as emitter model and reverb can change

Function	Description
<i>IRSX</i>	Manages a 3-D sound environment. This is the primary interface for managing a 3-D sound environment.
<i>IRSXEmitter</i>	Controls the media behavior, position, and orientation of the emitters in an audio environment. This abstract base class defines the general characteristics of audio emitters. Both <i>IRSXCachedEmitter</i> and <i>IRSXStreamingEmitter</i> provide this functionality through their interfaces.
<i>IRSXCachedEmitter</i>	Provides methods for handling file, memory, or application resources for cached emitters.
<i>IRSXStreamingEmitter</i>	Provides methods for handling real-time audio data for stream-based emitters.
<i>IRSXListener</i>	Controls the position and orientation of a listener in the audio environment. This abstract base class defines the general characteristics of audio listeners. Both <i>IRSXDirectListener</i> and <i>IRSXStreamingListener</i> provide this functionality through their interfaces.
<i>IRSXDirectListener</i>	Provides methods for connecting and disconnecting a direct listener from an output device.
<i>IRSXStreamingListener</i>	Provides methods to request the synchronous generation of audio output for streaming listeners.

Table 1: RSX interfaces.

abruptly without adverse effect and therefore do not need to be interpolated.

Keys are used to perform the correct action at a specific time. Each key only knows how to do one thing—*Run()*—and each has an *m_fTime* associated with it. The idea is that a large collection of them can be hung in a linked list, then iterated through and executed at the proper time. When WaveSpinner calls *SetTime()* on the emitter object, for example, the emitter walks through its linked list of keys and calls *Run()* on every key whose *m_fTime* value is between the previous time and the current time.

I use the *CSplineArray* class to handle interpolation. To refresh your memory, a spline is an algorithm that allows you to smoothly interpolate a curve between a set of points. The *CSplineArray* class encapsulates a one-dimensional spline, and you can combine these to handle splines of higher dimensions. I have thoughtfully provided a *3DSpline* class that contains—you guessed it—three instances of the *CSplineArray* class. The 3-D spline is used for position and orientation of emitters. Orientation of the listener uses a 6-D spline set, since a listener requires a 3-D up direction as well as a 3-D forward direction in order to fully specify orientation. Combining the dimensions this way is possible because you

can interpolate a point in *n*-dimensional space by interpolating each of the dimensions independently (I don't know if this is a *theoretically* correct statement, but it does seem to work in practice).

I appropriated (stole) the spline algorithm from *Numerical Recipes in C*, Second Edition, by William H. Press and others (Cambridge University Press, 1996). I chose this

Doppler is used to simulate sounds in relative motion

implementation because it is a 2-D spline, as I will be feeding it several 2-element values, namely time and a value. Because the times will not be on nice integer boundaries, I needed a spline that could handle several widely scattered points. For example, the user may give me points like (0.0, 2.0), (0.25, 1.0), and (3.75, 0.0). With the spline algorithm I chose, I can hand in an arbitrary time and it will magically figure out what the output value should be.

Upon examining the spline code, you will see that it closely follows the code in the book. I have added code to handle some special cases and boundary conditions. Note that before the spline can be used, some tables need to be set up that contain all the input points. The information for these tables was gleaned from the information passed in through the *ParseCommand()* method.

Final Notes

Note that to build WaveSpinner, you need to have RSX installed (available at <http://www.intel.com/ial/rsx/>). A good addition to WaveSpinner would be a GUI interface that lets you drag and drop sounds into position and plot their animation sequence. Writing the script by hand is quicker than writing a program to animate the sounds, but still becomes tedious. Another enhancement would be to add animation loops. Currently, if you want a sound to rotate every five seconds for a period of 30 seconds, for example, you would need to repeat the orientation sequence six times in the script. It would be easier to enter it once and have WaveSpinner take care of the rest.

DDJ

(Listings begin on page 89.)

REVENGE OF THE NURBS

L. PIEGL, University of South Florida, Tampa, and W. TILLER, GeomWare Inc., Tyler, TX

The NURBS Book

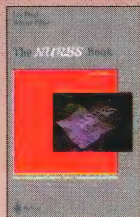
Second Edition

"...I highly recommend the book to anyone who is interested in a detailed description of NURBS. It is extremely helpful for students, teachers, and designers of geometric modelling systems." — COMPUTER APPLIED DESIGN

"It is literally a beautiful book...the hundreds of figures (all computer drawn) are both beautiful and pedagogically effective...The authors provide a wealth of detailed, worked-out examples of nearly every concept, not shunning complex calculations..." — COMPUTING REVIEWS

1996 • approx. 650 pp., 334 illus. • Softcover • \$54.95
ISBN 3-540-61545-8

Monographs in Visual Communications

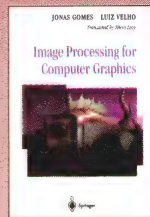


J. GOMES, and L. VELHO, both Instituto de Matematica Pura e Aplicada, Rio de Janeiro, Brazil

Image Processing for Computer Graphics

Concentrates on describing and analyzing the underlying concepts rather than on presenting algorithms or pseudocode. Covers: signal processing, data discretization, the theory of color and different color systems, operations in images, dithering and half-toning, warping and morphing, and image processing.

1997 • approx. 352 pp. • Hardcover • \$44.95
ISBN 0-387-94854-6



G. SEED, Heriot-Watt University, UK

An Introduction to Object-Oriented Programming in C++

With Applications in Computer Graphics

Helps readers develop good programming technique while demonstrating when and how to use more advanced features. Covers all major concepts, including templates, exception handling, RTTI, plus recent additions to the standard: namespaces, explicit, mutable and typename. Also features tools for graphics programming, and all program files (over 500) are available via the Internet. No prior knowledge of C or C++ is required

1996 • 1048 pp. • Softcover • \$54.50
ISBN 3-540-76042-3

R. ROBSON, Mississauga, ON, Canada

Using the STL

The C++ Standard Template Library

Introduces the STL to readers already familiar with C++. Presents a detailed discussion of the capabilities of the STL, the use of containers, descriptions of the algorithms and how they can be used, and how to extend the STL. An appendix provides an alphabetical reference to the entire STL.

1997 • approx. 436 pp., 34 illus. • Softcover • \$44.95
ISBN 0-387-98204-3

J. COWELL, De Montfort University, Milton Keynes, UK

Essential Java Fast

How to Write Object-Oriented Software for the Internet

Skip the long-winded lectures. Get past the jargon. Just learn the basics and start programming. Learn all about objects, applets, and class, then build towards more advanced techniques like animation, live updating, and two-way interactions. With this book, you can create substantial Java applications after only a few chapters.

1997 • 186 pp. • Softcover • \$24.95
ISBN 3-540-76052-0

To Order

Visit: Fine bookstores near you.

Call: 1-800-SPRINGER

e-mail: orders@springer-ny.com

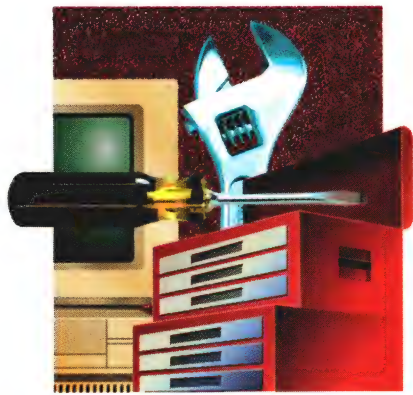


Springer

7/97

Reference: S973

CIRCLE NO. 236 ON READER SERVICE CARD



Tecate and Interactive 3-D

Combining networking and 3-D objects

Peter D. Kochevar

Browsers such as Netscape's Navigator or Microsoft's Internet Explorer use a text-based "document" metaphor to interface computers and the Internet. An alternative metaphor is to provide 3-D virtual worlds that let you conduct experiments in virtual laboratories, see how to replace a virtual part from the same vantage point a real part would have to be replaced, plan itineraries by previsiting 3-D virtual destinations before actually embarking on journeys, and more.

The Tecate system provides an infrastructure in which application-specific browsers can be quickly crafted to meet the demands of end users. Tecate provides a set of tools for describing animated 3-D scenes and their interaction with end users, which enables the description of interfaces to existing applications and data sources such as the World Wide Web, enables remote collaboration, and fosters software reuse through object orientation. Tecate's networking capabilities allow these worlds to interface with the Web, external databases, other worlds, and existing applications. The modular, object-oriented nature of Tecate worlds allows their components to be reused.

Tecate is publicly available at <ftp://ftp.sdsc.edu/pub/sdsc/graphics/tecate/tecate.tar.gz>. It runs on most UNIX platforms that support OpenGL 3-D. The Mesa OpenGL emulation library can be used for those systems that don't have OpenGL support. Tecate has been successfully installed on Digital Alpha and SGI Irix systems.

Tecate and Other Technologies

Tecate is not a browser—it is a collection of powerful components that can be combined to create application-specific browsers. Traditional browsers are really

"black boxes" that offer limited customization through the setting of parameters. (One exception to this architecture is Sun's HotJava browser, which adopted a component architecture similar to Tecate.)

Tecate uses virtual worlds as an underlying browsing metaphor instead of documents. A Tecate world can share information with other remote worlds. That is, a single world can be built in which multiple users can enter and independently navigate. This is something that is just beginning to appear in products such as Black Sun's CyberHub, a VRML-based multiuser server. The Tecate MOO example demonstrates this capability.

Tecate's API is the Abstract Visualization Language (AVL), a typeless scripting language based on Tcl. AVL extends Tcl by adding object-oriented programming support, 3-D animation, and a sophisticated event-handling mechanism. The growing use of scripting languages such as AVL is evidenced by the recent proliferation of JavaScript, VRMLscript, NeXT's WebScript, VBScript, and the like. AVL differs from many of these languages in that it has 3-D graphics incorporated within it, something that is just beginning to happen with the other languages. If interpreted AVL is too slow, Tecate allows time-critical behavior to be implemented in a compilable language such as C or C++.

AVL has taken a different approach than other 3-D graphics languages. In VRML, for instance, the stress is on creating 3-D graphics and not on specifying the behaviors of objects. VRML's behavior system is loosely coupled to a graphics system based on OpenInventor. In AVL, graphics and behaviors are more tightly coupled. AVL stresses object-object interactions, with graphics appearing as a side effect. In general, VRML is still rooted in classical computer graphics methodology—the hierarchical scene graph—while AVL takes a more modern, object-oriented approach.

Although both Tecate and VRML support interactive 3-D graphics, the two systems are very different. VRML uses a traditional "scene graph" to define the graphics, with behavior defined separately. Tecate provides a modern, object-oriented environment in which the graphics are a side effect of object-object interactions. Tecate's full-featured programming language supports networking and database queries, making it easier to dynamically access and visualize information from within Tecate worlds than from within their VRML counterparts.

Tecate's Architecture

Tecate is comprised of a run-time system and API (see Figure 1). The run-time system is an extensible, object-oriented system whose major components are objects as well as entities that appear within interactive virtual worlds. All objects possess four classes of properties:

- **Appearance.** Attributes that affect an object's visual appearance such as geometric and topological structure, color, texture, material properties, and so on.
- **Behaviors.** A set of procedures invoked upon receipt of messages from other objects.
- **State.** A collection of variables whose values represent an object's state.
- **Subobjects.** A list of objects that are parts of a given object, just as a wheel is a part of a car.

Tecate's run-time system has a layered architecture consisting of a kernel, a complement of system services, and a toolkit of visualization and user-interface components (see Figure 2). The kernel consists of the Object Manager, the Rendering Engine, and the Communications Stub. The Object Manager manages the creation, deletion, modification, and communication of objects. The Rendering Engine is responsible for creating visual and aural

Peter, a software engineer with Digital Equipment Corp. is in residence at the San Diego Supercomputer Center. He can be contacted at kochevar@sdsc.edu.

Network software that's well-designed, well-implemented and well-reviewed.

**"Best enterprise management
software of 1996."**

*– Infoworld praises Solstice™ Enterprise Manager™ software**

**"Solaris is a highly scalable, powerful and flexible
OS well suited to general Internet
service and enterprisewide intranet services."**

– Network Computing raves about Sun™ Solaris™ operating environment†

**"Many new Java™ development tools were released
this year, but none as radical as Sun's Java™ Workshop.™"**

– PC Magazine honors Java WorkShop Software‡

**And there's a whole lot more where that came from.
To see for yourself, visit our website or give us a call.**

Develop, deploy and manage with Sun™ Workshop,™ Solaris, and Solstice™ Software.

www.sun.com/sunsoft/reviews/ or contact 1-800-SUNSOFT (786-7638)



THE NETWORK IS THE COMPUTER™

*January 27, 1997, †February 1, 1997, ‡Reprinted from PC Magazine December 17, 1996-Copyright (c) Ziff-Davis Publishing Company 1996, © 1997 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun Logo, Solstice Enterprise Manager, Solstice, Sun Workshop, SunSoft, Solaris, Java, Java Workshop and The Network Is The Computer are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

CIRCLE NO. 362 ON READER SERVICE CARD

Visit the Dr. Dobb's Web site!

www.ddj.com

Featured Sections:

Subscription Services
Need to change your address, subscribe, renew your subscription or miss a recent issue, contact our subscription department with our on-line subscription forms.

Dr. Dobb's Index
Subject and author index of all the articles published by Dr. Dobb's dating back to 1982.

Dr. Dobb's Source Code
Convenient, quick access to all the Dr. Dobb's source code, dating back to 1991.

Dr. Dobb's Software Careers
Over 100 jobs, from the leading software development companies.

Dr. Dobb's Product Locator
Need to find a product or service, connect to our Product Locator and do a quick search...over 1,000 listings.

Plus:

- Selected articles
- Table of Contents of each Dr. Dobb's dating back to 1993
- On-line information, screen shots, and ordering for the Dr. Dobb's CD-ROM Library
- Author submission guidelines
- And much, much, more!

www.ddj.com

(continued from page 72)

renditions of virtual worlds that represent data. The Communications Stub provides the means for multiple Tecate sessions to run simultaneously while sharing a single virtual environment, thus fostering collaboration among end users.

Tecate's system services consist of interfaces to the Web and to databases managed by the Illustra database management system. The web interface allows any Tecate object to specify a URL from which information is to be drawn, and then mapped into an appropriate virtual world (Figure 3). The database interface functions similarly except that Tecate objects can pose queries written in SQL and the query results are then mapped into an animated, 3-D scene that represents the informational content of the queried data (Figure 4).

You can add new system services to Tecate by providing a set of functions that can be invoked by other Tecate objects. These functions correspond to the behaviors that are called when the service receives a message from other objects. Tools are provided to register the behav-

iors with Tecate and to manage the communication between the object providing the service and other Tecate objects.

Tecate's toolkit is a set of predefined objects that can be used for developing applications. The toolkit contains windows, lights, and cameras used to illuminate and render virtual worlds on a computer display. The toolkit also contains a collection of 3-D UI widgets—sliders, menus, icons, legends, coordinate axes, and the like—that can be used within virtual worlds. Finally, the toolkit contains a clock that is used to pace animations and trigger events. As with the system services, Tecate's base toolkit can be augmented by application programmers.

Describing 3-D Animation

AVL applications are descriptions of active objects. Tecate focuses on creating these objects and defining how they interact with one another over time. Although these objects can have geometry and appearance properties, the handling of graphics is not a primary concern of Tecate. Rather, 3-D animation is a side effect of general object-to-object interactions.

A collection of objects defined within Tecate constitutes a virtual world. Objects in a virtual world are "active" because each object functions independently from other objects under a separate thread of control. Objects interact with one another by passing messages between them. When objects receive messages, handlers are invoked to respond to the messages. These handlers are methods, called "behaviors," that are encapsulated within the objects. Within behaviors, objects can send new messages to other objects; create new objects or destroy old ones; and objects can alter the state, appearance, behavior, and subobject properties of either themselves or other existing objects.

Once all of the objects have been defined, a virtual world is set in motion by

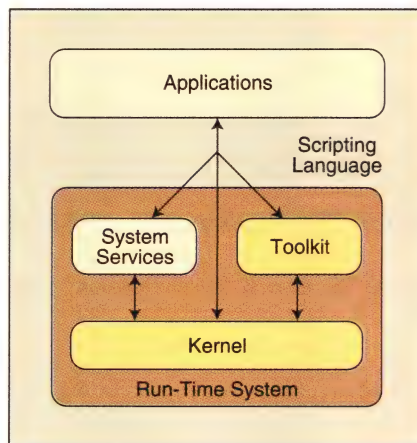


Figure 1: The Tecate system and its relationship to application programs.

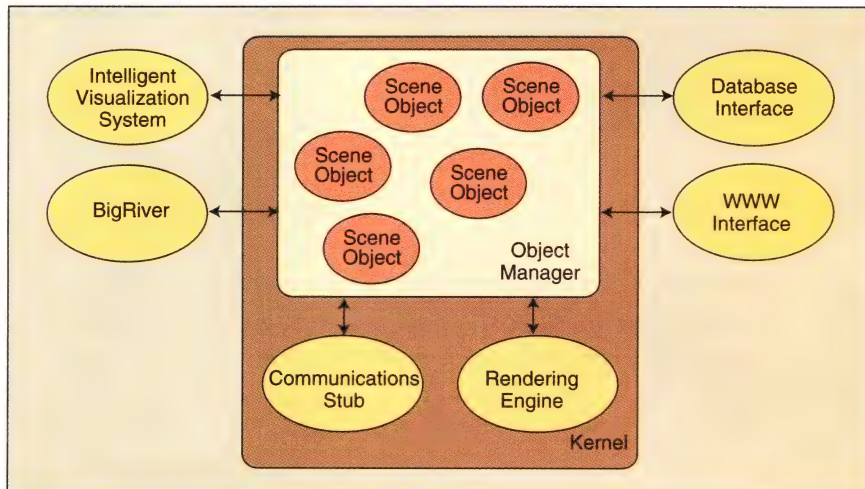


Figure 2: Detail of Tecate's kernel along with the system services provided by Tecate.

sending messages to those objects. These objects process the messages, usually resulting in a new set of messages being sent to yet other objects and so on. Often messages are sent to specific objects by users via input devices such as a mouse or keyboard. In Tecate, input devices are objects that send messages to other objects whenever input is provided. Objects wishing to be informed of input events register themselves with the associated input device objects.

Not all input devices trigger solely upon direct user interaction. For instance, within Tecate, clocks can be set under program control. At every clock tick, all objects that have registered with a clock are sent a message informing objects of the tick. These clocks can be used to time events or trigger actions. More importantly, the clocks are used to pace animations, allowing objects to periodically update their state and appearance.

Aside from object-object relationships via inheritance, AVL also supports object-subobject relationships, which bind one object to another. Currently, there are two types of constraints supported within Tecate:

- **Logical Constraint.** An object is logically bound to another; there is no formal geometric relationship between the two objects. Logical constraints are useful for creating sets of objects.
- **Transformational Constraint.** One object is constrained positionally, orientationally, and/or size-wise to another object. Transformational constraints are used for positioning objects in a scene and for creating part/subpart relationships. Note that for an object to appear within a scene, it must be transformationally constrained to the scene object or to some object that can trace transformational constraints through part/subpart relationships all the way to the scene object.

An object can be logically constrained to any number of objects while at the same time being transformationally constrained to at most one other object. Also, an object can have any number of sub-objects of either constraint type provided that all the subobjects are distinct.

In Tecate, there is a single object-creation operation called "cloning," and any object in the system can serve as a prototype from which a copy can be made through the clone operation. A clone inherits properties by maintaining a link to its prototype. When a reference to a property is made within an object, the system looks for the property value locally within the object. If no property value is found, then the object's prototype is searched in order to associate a value with the reference. If

NEW HIGH SPEED RISC I/O

Call us today at telephone 519/836/1291 or
facsimile 519/836/4878 to find out more about
the Intellicon-NT960 and other Connect Tech
communication products



Connect Tech Inc.

'Meeting your objectives...by design'

727 Speedvale Ave. W.
Guelph, ON N1K 1E6

Tel: 519-836-1291
Fax: 519-836-4878

Intellicon-NT960

a RISC based multi-port
I/O subsystem for
serial communications

Performance

- ★ An on-board Intel i960 32 bit RISC processor off-loads the serial I/O task from the main CPU
- ★ Offers up to 2MB of on-board dynamic RAM for data storage
- ★ Offers up to 256KB of on-board dual-ported RAM and 512KB Flash EEPROM for data storage and custom application programs
- ★ RISC-like quad UARTS with 24 bytes of FIFO per channel provide high speed data communications up to 115K baud on each port

Hardware Flexibility

- ★ The Intellicon-NT960 subsystem includes a NT960 host adapter, the ACM/16 external communications module, software drivers and manuals
- ★ Provides 16 to 128 asynchronous serial ports from one PC slot
- ★ Accommodates a total

- of 512 serial ports with 4 host adapters in a system
- ★ The ACM/16 module has both RJ45 and DB25 connectors
- ★ Offers optional on-board realtime clock and optional on-board battery backup for the dual-ported RAM
- ★ Supports 80286, 80386 and 80486 AT bus architecture

Software Flexibility

- ★ Development tools available for custom development
- ★ Software support for UNIX, XENIX, QNX, DOS

Reliability

Since 1985 Connect Tech's products have become known for their unsurpassed quality and reliability. This tradition of providing flexible cost effective solutions is based on Connect Tech's expertise in the design and manufacture of communication hardware and software. At Connect Tech we meet your objectives - by design.

CIRCLE NO. 131 ON READER SERVICE CARD

WARNING: Programmers in general have determined that without Opus Make your company's health may be seriously jeopardized.

In today's fast-paced world, time to market is everything. So why risk your company's health using a slow and unreliable build utility? You need Opus Make. It is the fastest, most full-featured, most reliable make utility on the market. Opus Make is designed to efficiently speed up your build process. You will develop your software faster, get it to market sooner and, in return, improve your company's health - and yours. Find out why Andrew Binstock says "Opus Make is the most compelling reason for not using the make utilities bundled with today's compilers."

Opus Make version 6.1 features include:

Makefile Compatibility: Processes Intersolv Configuration Builder™, Microsoft™ NMake and VC++, Borland Make™, and other makefiles.

Version Control Support: Access source files in Microsoft SourceSafe™, Burton Systems TLIB™, Intersolv PVCST™, and MKS Source Integrity™ version control systems.

Microsoft Visual C++ Support: Opus Make integrates with VC++ 2.x and 4.x.



Multiple Platform Support: DOS, OS/2, WinNT, Win95, AIX, HP/UX, SCO, SunOS, Solaris, Irix, DEC AlphaNT, and others.

Other features: Multiple-directory support • Pattern-based inference rules • Multiple targets created from single source • Automatic and in-line response files • Queued shell lines • Object library maintenance • Conditional, looping and include directives • And much more!

Opus MKMF V6.1 included: MKMF is our makefile and dependency generator. It quickly builds and maintains your makefiles. It understands C-preprocessor directives and resource compiler files. If you hate building makefiles by hand this is the tool for you!

Opus Software, Inc. • 1032 Irving Street, Suite 439 • San Francisco, CA 94122
Phone: (415) 485-9703 or 800-248-6787 • Fax: (415) 485-9704
Email: info@opussoftware.com • Web: http://www.opussoftware.com

Opus Make 6.1

NEW VERSION NOW AVAILABLE

VERSION 6.1

the prototype is itself a clone from some other object, then the prototype's prototype is recursively searched to resolve the reference. This type of "lazy" evaluation of property references is known as "delegation."

Note that subobject relationships are not inherited, that is, Tecate only does shallow cloning. Deep cloning, in which the subobjects are recursively cloned, must be done programatically within AVL. Subobject properties are not "intrinsic properties" of an object, as are state variables, appearance attributes, and behaviors. Only intrinsic properties are inherited.

With Tecate's delegation model, a change in a property value can affect all other objects cloned from the original object. This type of semantics is useful for establishing class-instance-like relationships between objects. For instance, one object may represent a particular class of automobile tire while all clones of the object would represent class instances. If a class-level change is needed that affects all instances (for example, a new tread pattern is to be introduced), only the object representing the tire class needs to change rather than having to change all instances individually.

The clone-prototype chaining implied by delegation can be overridden by changing the property values locally. For example, if one particular tire instance is to have a new tread pattern, then the pattern is altered in that instance only. References to the tread pattern for that object will use the local tread value rather than chain back to the tire class object. All other instances will continue to reference the value present in the tire class object.

Since AVL is a superset of Tcl, AVL possesses the standard program-control constructs such as if-then-else, switch, for-

loops, and so on. In addition, AVL has a rich set of list and string manipulation commands that it inherits from Tcl.

AVL adds several additional commands to the standard Tcl instruction set. The *clone* command is the object-creation command within AVL, and the *delete* command is the complementary operation to delete objects from the system. Object properties are specified and manipulated using the *add* command and deleted using the *remove* command. Behaviors in one object are initiated by another object using the *send* command, which specifies the behavior to invoke and the arguments to be passed. Queries about object properties can be made using the *inquire* command. The *which* command is used to determine where an object's properties are actually defined in light of Tecate's use of delegation to resolve property references. Finally, AVL provides a rich set of matrix and vector operators that are useful when positioning objects within 3-D scenes.

A 3-D MOO Example

To illustrate some of the capabilities of Tecate, I'll use AVL to build a framework for a 3-D Multi-User Dungeon Object-Oriented (MOO). A MOO is a shared virtual world that individuals can enter and act within, independent of other users. Prior to entering a shared world, each user defines an avatar—a 3-D model that will represent his or her presence in the world. As users move about the world, their avatars also move, allowing others to see who is present in the world and what they are looking at.

A 3-D MOO framework can be implemented as an AVL program. A copy of this program is run on each user's computer, creating a local copy of the shared world. This copy interacts with its counterparts

to ensure that the copies remain consistent over time.

The program also allows users to enter and leave the shared world. When new users enter a world, their copies of the program request a snapshot of the world's current state from some other copy. The new user's program then builds its local copy of the world based on the information that it is sent. In addition, the presence of new users is broadcast to all others in the MOO, so that their programs can begin sending updates to new users.

In the program, the object named *moos* (see Listing One, listings begin on page 90) is responsible for creating the local copy of the shared world and for maintaining state consistency. To simplify the example, the specification of the shared world is hardwired into the program. Furthermore, I assume that the shared world does not change over time save for the movement of participants' avatars. Therefore, the state of the MOO at a particular time can be described simply by listing the position and orientation of each avatar.

The *moos* object is cloned from the *WorldViewer* object, part of Tecate's basic toolkit. *WorldViewer* manages the display of the MOO world, and it provides a participant with helicopter-like flight controls for navigating the world. *WorldViewer* is an "abstract" object; it is not intended to be used directly but rather serves as a template from which other objects can be cloned.

To join a MOO session, new users send a bootstrap message to the *moos* object on their local host computer. The name of a remote host computer that is already running is passed as an argument, as is a specification of the shape of the avatar new users wish to assume. The *moos* object on the remote host is then informed that a new user wishes to join the MOO via the *enter* message. The remote host's *moos* object responds by sending the new user's local *moos* object the current state of the MOO using the *startUp* message. The remote host's *moos* object also informs all current MOO participants that new users are joining the session by broadcasting a *newPlayer* message. Note that in AVL, the *send* command can be used

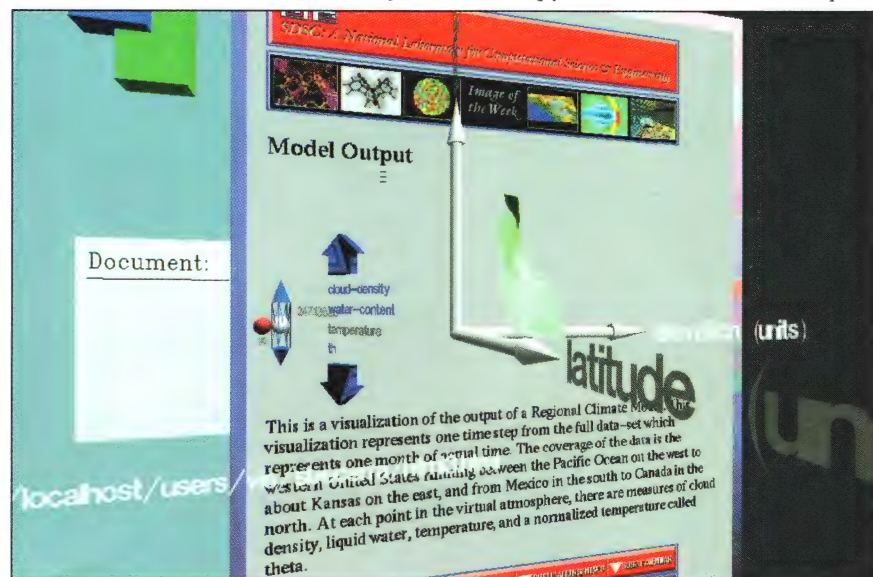


Figure 3: A visualization of a web page created using Tecate.

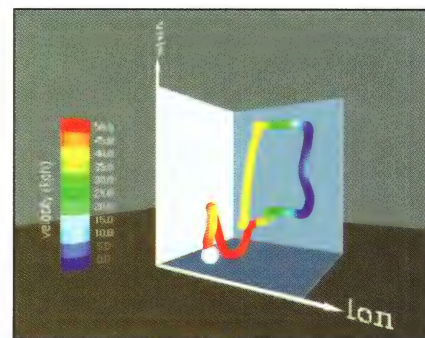


Figure 4: A visualization of data accessed via Tecate's database interface.

THE WAVE IS SPREADING!

C++ and Java Libraries and Builders

Planning to develop robust, scalable software applications? Rogue Wave is here to help you! Use our object-oriented C++ and Java products to develop for a wide variety of environments, including client-server, intranet and Internet environments. Get reliability and portability, and save time and effort using (and reusing!) our well-designed classes. Rogue Wave foundation class libraries and other software parts make it easy to add fundamental operations such as network and database connectivity, so you can focus on the core functionality of the application you're developing. Our libraries and builders are portable among Unix, Windows, and OS/2, and full source code is available, with no royalties or run-time fees.

Whether you're here in the US or overseas,
get the best in proven OO technology...contact us today!

In the United Kingdom:
Rogue Wave Software-UK.Ltd.
Phone: +44 118 9880224
Fax: +44 118 9880360
www.roguewave.co.uk

In The Netherlands:
Rogue Wave Software B.V.
Phone: +31 20-695 55 11
Fax: +31 20-695 17 61
www.roguewave.nl

Corporate Headquarters:
Rogue Wave Software
Phone: (800) 487-3217 or (541) 754-3010 Fax: (541) 757-6650
www.roguewave.com

In Germany:
Rogue Wave Software GmbH
Phone: +49 (61 03) 59 34-0
Fax: +49 (61 03) 3 69 55
www.roguewave.de

In France:
Rogue Wave Software S.A.R.L.
Phone: +33-1-47695803
Fax: +33-1-47695802
www.roguewave.fr

**Rogue
Wave
Software,**

a leading provider of world-class software parts for C++ and Java™ developers, has gone global! Now with four new European offices, we're able to offer you better service here and abroad. Call us for expanded local technical support and sales information, and quick access to our broad line of reusable, cross-platform C++ and Java libraries and builders.

The Software Parts Company™



**Rogue
Wave**
SOFTWARE

The Fastest Way To The NT™ DDK



When you're developing mission critical device drivers,
only the best of breed tools will do!

Version 2.0 is here!

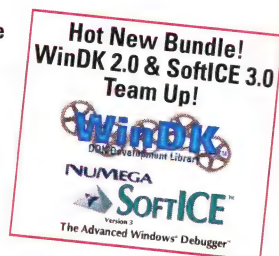
- Over 11,000 lines of library source code
- Over 19,000 lines of samples on CD



See What the Coolest Device
Driver Wizard Can Do!

Free Download at:
www.bluewatersystems.com

- Shipping USB/WDM support
- Kernel mode profiling
- 500 page manual



800.962.2114

Tel 425.771.3610 • Fax 425.771.2742
E-Mail: info@bluewatersystems.com
NO RUNTIME ROYALTIES
VISA, MC, AMEX & Approved P.O. accepted

CIRCLE NO. 180 ON READER SERVICE CARD

Check our web site
for other great tools!



The Best Java ORB

Visigenic's VisiBroker The ORB of Choice

Why are leading software companies turning to VisiBroker for Java and VisiBroker for C++ to implement IIOP in their product sets? It's simple. Evaluations have demonstrated VisiBroker's superior interoperability and scalability. And since Visigenic is the clear leader in Java ORB technology—having produced the first Java ORB and the first implementation of IIOP, VisiBroker is the logical choice.



Call Visigenic at

1-800-800-0361 or
1-415-312-7197
e-mail info@visigenic.com.

To download a Free evaluation copy of VisiBroker for Java
or VisiBroker for C++ please visit our web site at
www.visigenic.com

CIRCLE NO. 361 ON READER SERVICE CARD

(continued from page 76)

to send a message to an object residing on a remote machine by prepending a host name to the name of the object that is to receive the message.

For simplicity, the name of a distinguished host computer is hardwired into the single program that all MOO participants run. That host must be the first to run the program, thus providing all other participants' programs with a source for acquiring the current state of the MOO.

The *move* behavior in *moo* (inherited from *WorldViewer*) is responsible for altering a user's position within the shared world of the MOO. This behavior is linked automatically to certain mouse and keyboard events when the *init* message that is inherited from *WorldViewer* is sent to the *moo* object. To the default *move* behavior, *moo* adds a broadcast of a user's position and orientation to all other MOO participants.

In this example, the shared world of the MOO consists of a collection of rooms. Each room is a clone of the *Room* abstract prototype object; see Listing Two. Unlike rooms in a building, the rooms within the implemented MOO are not spatially adjacent to one another. Instead, each room serves as a virtual world unto itself. At any given time, a user is located in a single room, and since all rooms in the MOO have no windows, a user can only see the contents of the room in which he or she currently resides. Therefore, only one room needs to be displayed at any given time. When users are in a room, that room is considered to be "realized," and all other rooms are "unrealized."

MOO participants can move from one room to another by crossing a "bridge." A bridge consists of two "portals," each of which appears as a disk in the rooms spanned by the bridge. Figure 5 is a typical room in the MOO containing a few portals, and two avatars signifying the presence of other users in the room. Listing Three presents code for the *Bridge* abstract prototype object (code for a *Portal* abstract prototype object is available electronically; see "Availability," page 3).



Figure 5: A user's view of a room within a MOO. Disks represent portals. The two other objects are avatars.

To travel from one room to another (a process called "wormholing"), a MOO participant can perform one of two actions—fly into a disk representing a portal, or select a disk with the mouse and be drawn automatically into a portal. Both mechanisms rely on the *proximity* behavior within a *Portal* object. If a user moves within a certain predefined distance from the center of a portal, the user is then wormholed into another room by the *wormhole* behavior of *Portal*.

Proximity testing is performed each time users move within a realized room. In the body of the *init* behavior, portal objects notify the room objects in which they reside that they wish to be informed of "realize" events. Then, when a room is first redisplayed, a portal's *enableProximity* behavior is invoked. This behavior registers with the viewer object that a portal is to be informed of "move" events. When so registered, user movement automatically causes a portal's *proximity* behavior to be invoked.

When users pick a disk representing a portal, they are automatically flown into the portal. The process by which this happens illustrates how animation occurs within Tecate: Objects wishing to animate inform a clock object that they want to receive clock tick events. In the MOO program, a mouse click on a portal causes the portal to register itself with the predefined clock object called *SystemClock*. At each clock pulse, a portal's *tractor* method is called, which moves the user a tenth of the distance from their current position to the center of the portal.

Listing Four (available electronically) is the code segment where the MOO's shared world is defined, and all object initializations take place. The shared world consists of three rooms: *a*, *b*, and *c*. Bridges are created to allow users to wormhole from any room to any other room. Users are initially placed into room *a* when they first enter the shared world of the MOO. The name of a currently running host and a user's avatar shape description are assumed to be defined in the AVL variables *bootstrapHost* and *shape*, respectively. If the program is run on the host named in *bootstrapHost*, then the *mo* object there allows the user to move about the three-room world while waiting for entry requests from other users at remote sites.

Acknowledgments

This work was supported by the Digital Equipment Corp. and the San Diego Supercomputer Center. Further support was provided by a grant from NASA's Earth Observing System, Distributed Information System (EOSDIS) project. Thanks to Santi Becerra and Ryan Camoras for their help in constructing the latest Tecate prototype.

DDJ

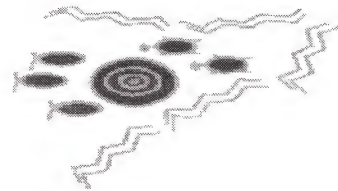
(Listings begin on page 90.)

DinkumwareTM, Ltd.

Genuine Software

billabong

A waterhole isolated from the main channel of a river. Less turbulent than the main stream, it provides a predictable environment for a variety of creatures.



Dinkum[®]EC++ Library Dinkum EST Library

A proper subset of the *Standard C++ Library* especially well suited for writing efficient code for embedded systems.

- ≈ supports `cin`, `cout` and `iostreams` classes
- ≈ includes class `string` and complex arithmetic
- ≈ is portable and thread safe

Visit our Web site: www.dinkumware.com

CIRCLE NO. 348 ON READER SERVICE CARD



SERIOUS Copy-Protection and Software Licensing with No Hardware Key and No Disk Key

With CrypKey, you not only stop unauthorized use and duplication of your software, you can **increase your sales!** CrypKey truly **adds** features to your software. Offer usage by runs, time, enable or add options by phone, fax or e-mail! With 27 function calls, CrypKey offers you complete flexibility.

Also available as CrypKey INSTANT

Protects 16 or 32 bit Windows app's in 5 minutes, with no coding! Just pick your protection options, like Ready-To-Try, click the EXE you want to protect, and your software is ready to sell itself from your website, CD-ROMs or floppies with complete security!

Ready-To-Try feature offers secure Web distribution of your software!

Allows 1 trial period, like 30 days or 5 runs. Back-dating or even deleting the directory and re-installing doesn't fool CrypKey. Your first customer contact is when they buy!

YES, It Really Works!

Professional programmer evaluations agree: CrypKey offers the best combination of security, reliability, economy and ease of use with near-zero support. Supports Win3.x/95/32s/NT/95B(FAT-32), OS/2, DOS and intelligently manages network licensing on all Novell & Microsoft based networks.

100% Satisfaction Guaranteed!

Kenonic Controls Ltd., Calgary, Canada, Phone: 403-258-6200 Fax: 403-258-6201 e-mail: crypkey@kenonic.com

CrypKey INSTANT is **FREE on the Web**, Ready-To-Try for 30 days, with full security, just like your software can be! <http://www.kenonic.com/crypkey.htm>

CIRCLE NO. 310 ON READER SERVICE CARD

NEW! version 7.0

LEADTOOLS

OVER 40 FILE FORMATS

TIFF 6.0	CAL
PCX	DIB
BMP	EPS
PNG	FAX (raw)
KODAK	DCX
-FlashPix	IMG
-Photo CD	IOCA
JPEG	MODCA
-JFIF	MAC
-JTIF	MSP
-JPG	WINFAX
-Progressive	VDA
CMP	RAS
-Progressive	WMF
GIF	PSD
-Animated	WPG
-Interlace	TGA
-Transparency	PCT
-Embedded Text	

For a Complete List Visit
www.leadtools.com/format.htm



Auto Deskew

Boolean & Math Combine
Brightness & Contrast
Color Reduction & Expansion
Color Separation
Edge Detection
Emboss
Gamma Correction
Geometric Transforms
Halftone & Grayscale
Histogram Equalization
Hue & Saturation
Morphological Filters
Noise Removal
Region Of Interest
Sharpen & Blur
Spatial Filters
Watermarks

For a Complete List Visit
www.leadtools.com/processing.htm

PROCESS IMAGES

Animation
Auto dithering
Bitonal scaling options
Special paint effects
Zooming and Scrolling

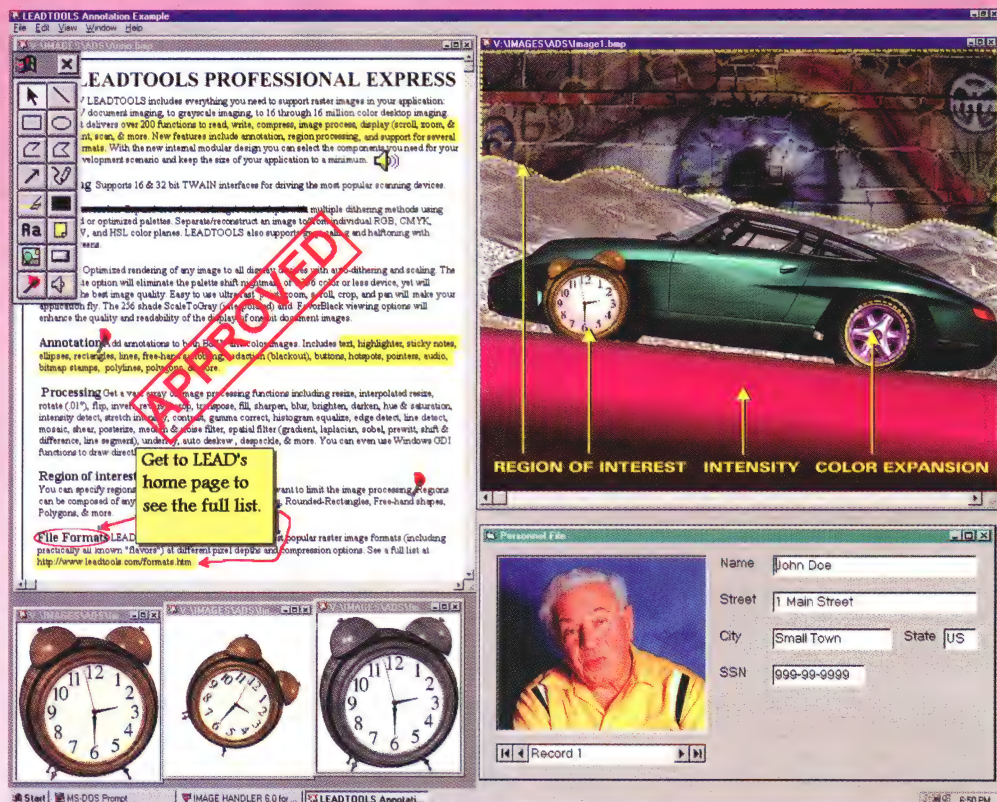
DISPLAY SUPPORT

Internet Aware
ActiveX control

AND MORE!

Scanning
Compression
Annotation
Database
Printing

LEADing technology in image development tools.



Microsoft uses LEADTOOLS to make FrontPage more powerful. Symantec/Delrina uses LEADTOOLS to make WinFax more powerful. Corel uses LEADTOOLS to make CorelDraw more powerful. Detect a trend? These companies and thousands of others use LEADTOOLS to make sure that their applications contain the best imaging technology available. These industry leaders know that they can trust LEADTOOLS to deliver the imaging technology they require in an easy to use, reliable and time saving professional development toolkit.

Whether you are working with document or medical imaging, creating an imaging database, or adding images to a web page, using LEADTOOLS ensures that your application has the highest level of imaging technology previously found only in professional applications like Photoshop. With LEADTOOLS you will get the job done quickly and easily.

LEADTOOLS is a comprehensive imaging toolkit with more than 200 functions, properties and methods, that help programmers integrate black & white, grayscale and color images into their applications. LEADTOOLS contains 11 categories include scanning, color conversion, display, annotations, image processing with region of interest, animation, compression, image format import/export filters, internet, database, and printing.

The award winning LEADTOOLS is available as 16 & 32 bit DLLs, 16 & 32 ActiveXs, and a VBX, and includes extensive example source code for C/C++, MFC, Visual Basic, Delphi, Visual FoxPro, and Access (2.0 & 7.0). LEADTOOLS' mature and stable code is backed up by a 30-day money-back guarantee (US & Canada only) and FREE technical support is available via phone, fax, BBS, Internet or CompuServe.

"If I had to write my own personal version of Photoshop, LEADTOOLS is where I'd start."

Rick Grehn
Byte Magazine

"LEADTOOLS is an indispensable tool, we used it in the development of our FrontPage application".

Tom Button
Director of Marketing, Internet Platform & Tools division
Microsoft



800-637-1837

900 Baxter St. Charlotte, NC 28204 704-332-5532 Fax: 704-372-8161 CompuServe: "GO LEADTECH"

LEADTOOLS is available in several versions, not all features are available in all versions.
*License required from Unifs for formats using LZW compression. LEAD and LEADTOOLS are registered trademarks of LEAD Technologies, Inc. All other product names are trademarks of their respective owners.

CIRCLE NO. 90 ON READER SERVICE CARD



<http://www.leadtools.com/>

IMPLICIT SURFACES

Listing One

```

LSurface Components: // Given Instance in Object Database
// Pre-Allocated Temporaries for Calculations
static float x1, y1, z1, x2, y2, z2, Tmp1, Tmp2, Tmp3, Tmp4, Tmp5, Tmp6,
                                          Tmp7, Tmp8, Tmp9;

float LSurface::MGSM(float x, float y, float z)
{
    // Initialization of Threshold and Number of Components
    float Total=Components.GetSize();
    int N=Components.GetSize();
    // Loop through all components
    while(N--) {
        Ci=&Components[N]; // Get Pointer to Current Component
        // Check For Any Angular Rotation
        if( !((Ci->Theta_X==0) && (Ci->Theta_Y==0) && (Ci->Theta_Z==0)))
        {
            // There is Angular Rotation So Use Angular Model

            // Perform Translation of Component
            Tmp1=x+Ci->Tx;
            Tmp2=y+Ci->Ty;
            Tmp3=z+Ci->Tz;

            // Calculate Temporaries for Rotation
            Tmp4=Tmp2*Ci->cos_z;
            Tmp5=Tmp1*Ci->sin_z;
            Tmp6=Ci->cos_z*Tmp1;
            Tmp7=Ci->sin_z*Tmp2;
            Tmp8=Ci->cos_y*Tmp3;

            // Perform Rotation
            x1=(Ci->cos_y*(Tmp6-Tmp7)+Ci->sin_y*Tmp3);
            y1=(Ci->sin_x*(Ci->sin_y*Tmp6- Ci->sin_y*Tmp7-Tmp8)
              + Ci->cos_x*(Tmp5+Tmp4));
            z1=(Ci->cos_x*(-Ci->sin_y*Tmp6+Ci->sin_y*Tmp7+Tmp8)
              + Ci->sin_x*(Tmp5+Tmp4));
            // Perform Stretching
            x2=x1*Ci->Sx;
            y2=y1*Ci->Sy;
            z2=z1*Ci->Sz;
            // Calculate Divisor
            Tmp9=(x2*x2+y2*y2+z2*z2);
            Total-=( Ci->Strength / (Tmp9< 0.000001? 0.000001 : Tmp9));
        }
        else {
            // There is No Angular Rotation (x1), So Use Simple Form

            // Translate and Stretch at the Same Time
            x2=(x+Ci->Tx)*Ci->Sx;
            y2=(y+Ci->Ty)*Ci->Sy;
            z2=(z+Ci->Tz)*Ci->Sz;

            // Calculate Divisor
            Tmp9=(x2*x2+y2*y2+z2*z2);

            // Apply to Threshold
            Total-=(Ci->Strength / (Tmp9< 0.000001? 0.000001 : Tmp9));
        }
    };
    return Total;
}

```

IMAGE PROCESSING

Listing One

```

#include <SGIReader.h>
#include <Contraster.h>

#include <SGIWriter.h>

int main ( int argc, char *argv[] )
{
    SGIReader in( "test.rgb" );
    Contraster contrasted( &in, 1.3 );
    SGIWriter
        out( "out.rgb", (InputImage *)&contrasted, in.getArea(), UChar);
    return ( 0 );
}

```

Listing Two

```

// File: Contraster.h -- (c) 1996 Mayur Patel

#ifndef Contraster_CLASS
#define Contraster_CLASS

#include <InputImage.h>
#include <Type.h>

class Contraster : public InputImage {
public:
    Contraster( InputImage *pIn, float rLumFactor, float rGreyPoint = 0.5 );
    ~Contraster( void );
    int
        fillTile( ImageTile *pWriteHere );
protected:
    InputImage *_pHost;
    float _rLumFactor;
    float _rGrey;
};
#endif

```

Listing Three

```

// File: Contraster.C -- (c) 1996 Mayur Patel

#include <Contraster.h>
#include <assert.h>

Contraster::Contraster( InputImage *pIn, float rLumFactor, float rGrey )
: InputImage( Real )
{
    _pHost = pIn;
    if ( _pHost )
        _pHost->registerReference();
    _rLumFactor = rLumFactor;
    _rGrey = rGrey;
}

Contraster::~Contraster( void )
{
    if ( _pHost )
        _pHost->unregisterReference();
    return;
}

int
Contraster::fillTile( ImageTile *pWriteHere )
{
    int iRet = 0;
    ImageTile *pTile;
    unsigned long lLoop;
    register float *pSrc;
    register float *pDest;
    if ( _pHost && pWriteHere )
    {
        pTile = _pHost->newTile( pWriteHere->getArea() );
        if ( pTile )
        {
            assert( getType() == pWriteHere->getType() );
            pTile->typecast( Real );
            pSrc = (float *) pTile->getBuffer();
            pDest = (float *) pWriteHere->getBuffer();
            lLoop =
                pWriteHere->getArea().width *
                pWriteHere->getArea().height;
            while ( lLoop )
            {
                lLoop--;
                // contrast enhance:
                *pDest = _rGrey + ( ( *pSrc - _rGrey ) * _rLumFactor );
                // clamp over-exposure & under-exposure:
                *pDest = ( *pDest < 0.0 ) ? ( 0.0 ) : ( *pDest );
                *pDest = ( *pDest > 1.0 ) ? ( 1.0 ) : ( *pDest );
                pDest++;
                pSrc++;
            }
            pTile->deleteTile();
        }
    }
    return ( iRet );
}

```

Listing Four

```

#ifndef TYPESEPER
#define TYPESEPER ;
#endif

#ifndef TYPEENDOFLIST
#define TYPEENDOFLIST ;
#endif

TYPEMACRO( float, Float )
TYPESEPER
TYPEMACRO( int, Int )
TYPEENDOFLIST

#ifndef TYPESEPER
#define TYPESEPER
#endif

#ifndef TYPEENDOFLIST
#define TYPEENDOFLIST
#endif

#ifndef TYPEMACRO
#define TYPEMACRO
#endif

```

Listing Five

```

enum Type {

#define TYPESEPER ,
#define TYPEENDOFLIST
#define TYPEMACRO( type, label ) label
#include <Type.m>
};

```

Listing Six

```

void
addition( Type opType, void *pDest, void *pOp1, void *pOp2 )
{
    switch( opType )
    {
        #define TYPEMACRO( type, label ) \
        case( label ) : \
            *(( type *) pDest ) = \
                ( type ) *pOp1 + ( type ) *pOp2; \
            break;
    }
}

```

(continued on page 82)

(continued from page 81)

```
#include <Type.m>
default:
    break;
};
return;
}
```

RAVEKIT

Listing One

```
void Raver::TimeSlice(long keys)
{
    TQAVGouraud v1, v2, v3;
    TQAVTexture t1, t2, t3;
    TQATexture *tex = NULL;
    Tri3D *this_tri = tri_list;

    // The generic Renderer base class handles moving the 3D "world". The
    // vertex objects within the world take care of "projecting" the
    // visible vertices of that world onto a 2D viewport and the Tri3D
    // class contains the Gouraud shading and/or texture info

    // call base class, handle vertex transformations
    // according to input info in keys bitfield
    Renderer::MoveWorld(keys);

    QARenderStart (drawContext, NULL, cache);
    // sadly, no cache cap in default engine

    while (this_tri) {
        // In this example, clipping is only at triangle level. With a bit
        // of edge checking, the gaps could be filled in with a maximum of
        // 2-for-1 triangles (3-for-1 max on corners)
        // uv co-ords also need to be offset
        Vertex3D &v3a = vertex_list[this_tri->v1];
        Vertex3D &v3b = vertex_list[this_tri->v2];
        Vertex3D &v3c = vertex_list[this_tri->v3];
        if (this_tri->GetClip(vertex_list)) {
            if (this_tri->texmap > 0 &&
                this_tri->texmap < num_textures) {
                tex = texTable[this_tri->texmap].tp;
                if (tex) {
                    QASetPtr(drawContext,
                        kQATag_Texture, tex);
                    t1.x = v3a.pix_point.x;
                    t1.y = v3a.pix_point.y;
```

```
t1.z = v3a.invZ;
t1.invW = v3a.invZ;
t1.a = 1.0f;
t1.r = 1.0f;
t1.g = 1.0f;
t1.b = 1.0f;
t1.uOverW = v3a.u;
t1.vOverW = v3a.v;
t2.x = v3b.pix_point.x;
t2.y = v3b.pix_point.y;
t2.z = v3b.invZ;
t2.invW = v3b.invZ;
t2.a = 1.0f;
t2.r = 1.0f;
t2.g = 1.0f;
t2.b = 1.0f;
t2.uOverW = v3b.u;
t2.vOverW = v3b.v;

t3.x = v3c.pix_point.x;
t3.y = v3c.pix_point.y;
t3.z = v3c.invZ;
t3.invW = v3c.invZ;
t3.a = 1.0f;
t3.r = 1.0f;
t3.g = 1.0f;
t3.b = 1.0f;
t3.uOverW = v3c.u;
t3.vOverW = v3c.v;
QADrawTriTexture(drawContext, &t1, &t2,
    &t3, kQATriFlags_None);
}
else {
    v1.x = v3a.pix_point.x;
    v1.y = v3a.pix_point.y;
    v1.z = v3a.invZ;
    v1.invW = 1;
    v1.a = 1.0f;

    v2.x = v3b.pix_point.x;
    v2.y = v3b.pix_point.y;
    v2.z = v3b.invZ;
    v2.invW = 1;
    v2.a = 1.0f;

    v3.x = v3c.pix_point.x;
    v3.y = v3c.pix_point.y;
    v3.z = v3c.invZ;
    v3.invW = 1;
    v3.a = 1.0f;

    if (this_tri->type == TRI_MOVEABLE) {
        // calc dynamic lighting here
        // no-op for this demo
    }
    else {
        v1.r = this_tri->rgb1.r;
        v1.g = this_tri->rgb1.g;
        v1.b = this_tri->rgb1.b;

        v2.r = this_tri->rgb2.r;
        v2.g = this_tri->rgb2.g;
        v2.b = this_tri->rgb2.b;

        v3.r = this_tri->rgb3.r;
        v3.g = this_tri->rgb3.g;
        v3.b = this_tri->rgb3.b;
    }
    QADrawTriGouraud(drawContext, &v1, &v2, &v3, kQATriFlags_None);
}
this_tri = this_tri->next;
}
QARenderEnd (drawContext, NULL);
}
```



High Performance Imaging Annotation & Redlining Development Tools



RasterMaster 6.0 - Familiar API and well-proven technology in use world-wide by companies such as British Petroleum, Chase Manhattan, Eastman Kodak, Filenet, Ford Motor, Gannett, Hewlett-Packard, IMSI, Lexis, Philips, Polaroid, Siemens-Nixdorf, Trix, Unisys, Xerox and more.

50+ Raster Formats All TIFF, JPEG, Group3, Group4, MO:DCA, IOCA, CALS, PNG, PCX, GIF, PhotoCD, ABIC, **Flashpix** and more.

RasterNote Annotation/Redlining toolkit - Includes all popular features. Create Sticky Notes, Lines, Ellipses, Freehand Draw, Polygons and more.

NEW ADDITIONS: [RasterNote](#), [UNIX](#), [Alpha](#), [Flashpix](#)

Featuring:

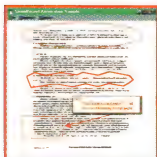
Format Reading, Saving, Compression, Display, Despeckle, Rotate, Zoom, Pan, Scroll, Deskew, Anti aliased display, Auto Aspect Ratio, Image processing, Transparent color, Improved Twain scanning, Color reduction, promotion and more

Platforms: Win 3.1, 95/NT, OS/2, Alpha, Macintosh, UNIX

Environments: DLL, VBX, OCX/ActiveX, Delphi, VB, PBuilder, VC++

CALL 617 630-9495 OR SEE OUR WEBSITE FOR **FREE, NO RISK EVALUATIONS!**

Imaging Software - Powerful, Fast, Reliable



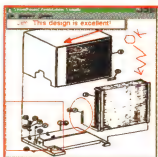
Snowbound Software

PO Box 520, Newton, MA 02159

FAX: 617 630-0210

www.snowbnd.com

Email: salesdd@snowbnd.com



CIRCLE NO. 309 ON READER SERVICE CARD

Listing Two

```
int Vector3D::Project()
{
    if (z <= view_plane) return -1000;
    invZ = 1.0f - view_plane / z;
    scaleZ = scaleZX / z;
    pix_point.x = (int)(raster_centerX + x * scaleZ + 0.5f);
    pix_point.y = (int)(raster_centerY + y * scaleZ + 0.5f);
    return ClipBounds();
}
```

Listing Three

```
// Renderer::MoveWorld() - called by derived class. Processes user input and
// sets up transformation & rotation values which it passes to TransformWorld()

void Renderer::TransformWorld(
    float tx, float ty, float tz,
    float rx, float ry, float rz)
{
    for (int n = 0; n < num_vertices; n++)
        vertex_list[n].Transform(tx, ty, tz,
            rx, ry, rz, T_PREROTATION | T_RELATIVE);
}
```

(continued on page 84)

**NEW:
C++ Object Classes**

Do you have a complex data modeling problem?



Don't compromise.

Other databases require design or performance concessions when implementing complex data models. The multi-model TITANIUM® database engine directly supports highly complex relationships, without redundant data, without intersect tables, without joins... without compromise.

Parts explosions, for example, and other many-to-many, recursive, and n-ary relationships are handled simply and elegantly by TITANIUM's patented Dynamic Pointer Array (DPA) technology.

Solve it with



**Optimal Performance for
Complex Development Needs**

TITANIUM's multi-model approach allows developers to use pure object database classes or straightforward navigational APIs for elegant code, while end-users work with simple relational and SQL statements for fast, usable queries.

A multi-threaded client/server and stand-alone database management system,

TITANIUM also features ODBC access, compiled stored procedures, fault-tolerance, transaction processing, on-line backup, and an extremely small memory footprint. All major PC and UNIX platforms are supported.

For a TITANIUM Technical White Paper, call 800-445-mdbs or 765-463-7200.

E-Mail: info@mdbs.com WWW: <http://www.mdbs.com>

Successful Applications Are Built With TITANIUM.

mdbs
Micro Data Base Systems, Inc.

mdbs and TITANIUM are registered trademarks of Micro Data Base Systems, Inc. ©1997 Micro Data Base Systems, Inc.

CIRCLE NO. 146 ON READER SERVICE CARD

(continued from page 82)

Listing Four

```
// Snippet from Vector3D::Transform()
// ...
else if (t & T_RELATIVE) {
    if (t & T_PREROTATION) {
        x += tx;
        y += ty;
        z += tz;
        Rotate(rx, ry, rz);
    }
}
// ...
// The rotate component of Transform(). Defined as inline
void Vector3D::Rotate(float rx, float ry, float rz)
{
    float sin_a, cos_a, temp;
    if (ry) {
        sin_a = sin(ry);
        cos_a = cos(ry);
        temp = x * cos_a - z * sin_a;
        z = x * sin_a + z * cos_a;
        x = temp;
    }
    if (rx) {
        sin_a = sin(rx);
        cos_a = cos(rx);
        temp = y * cos_a - z * sin_a;
        z = y * sin_a + z * cos_a;
        y = temp;
    }
    if (rz) {
        sin_a = sin(rz);
        cos_a = cos(rz);
        temp = x * cos_a - y * sin_a;
        y = x * sin_a + y * cos_a;
        x = temp;
    }
}
```

Listing Five

```
// Triangles are added to the linked list as indexes into the vertex array.
// Although RAVE has built-in support for meshes, I've kept it abstracted from
// the renderer. In Direct3D IM, for example, you can pass your own vertex
// index list with stride info and so avoid duplication of services
void Renderer::AddTri(
    int vi1, int vi2, int vi3, // vertex lookup indices
    Uv uv1, Uv uv2, Uv uv3, // texture coords
    Rgb &p_rgb, // the "center" color
    int tmp, int type) // texmap index, type "hint"
{
    Tri3D *active_tri, *new_tri;
    active_tri = tri_list;
    while (active_tri) {
        if (active_tri->next)
            active_tri = active_tri->next;
        else
            break;
    }
    new_tri = new Tri3D(
        vi1, vi2, vi3, uv1, uv2, uv3, p_rgb, tmp, type,
        lightList, vertex_list);
    if (!active_tri) // must be first one
        tri_list = new_tri;
    else
        active_tri->next = new_tri;
}
// Vertices are stored in an array for fast random lookup
int Renderer::AddVertex(
    float x, float y, float z,
    float r, float g, float b)
{
    if (num_vertices >= MAX_VERTICES)
        // allocate some more storage here
        // ...
        return -1;
    Vertex3D &vert = vertex_list[num_vertices++];
    vert.Install(x, y, z, r, g, b);
    return num_vertices - 1;
}
```

MOTION BLUR

Listing One

```
#define MAX_IMAGES
int motionBlur(char *firstImagePath, char *outputDir, int
    numFrames, int blurDepth)
{
    memImage *images[MAX_IMAGES];
    char directory[MAXPATH], fileName[MAXPATH], prefix[MAXPATH],
        inSuffix[MAXPATH];
    char currentPath[MAXPATH], inPath[MAXPATH];
    int frameNum, i, j, status;
    if (blurDepth > MAX_IMAGES) {
        statusPrint("motionBlur: blurDepth is larger than the limit.");
        return -1;
    }
    // the directory includes the drive letter
    status = getPathPieces(firstImagePath, directory, fileName, prefix,
        &frameNum, inSuffix);
    if (status != 0) {
        statusPrint("motionBlur: Check the first image pathname");
        return -2;
    }
    int imHeight, imWidth, bpp, frameCounter, row, col;
```

```
status = readBMPHeader(firstImagePath, &imHeight, &imWidth, &bpp);
if (status != 0) {
    sprintf(g_msgText, "motionBlur: Cannot open: %s", firstImagePath);
    statusPrint(g_msgText);
    return -3;
}
for (frameCounter = frameNum; frameCounter <= frameNum + numFrames; frameCounter++) {
    // Open and close the appropriate images
    if (frameCounter == frameNum) {
        for (i = 0; i < blurDepth; i++) {
            makePath(currentPath, directory, prefix,
                frameCounter + i, inSuffix);
            images[i] =
                new memImage(currentPath, 0, 0, RANDOM,
                    'R', RGBCOLOR);
            if (!images[i]->isValid()) {
                sprintf(g_msgText,
                    "motionBlur: unable to open image: %s",
                    currentPath);
                statusPrint(g_msgText);
                return -4;
            }
        }
    }
    else {
        delete images[0]; //close oldest image
        for (j = 0; j < numFrames; j++)
            images[j] = images[j+1]; //open new image
        makePath(currentPath, directory, prefix,
            frameCounter + blurDepth - 1, inSuffix);
        images[blurDepth - 1] = new memImage(currentPath, 0,
            0, RANDOM, 'R', RGBCOLOR);
    }
    // blur the images
    char outPath[MAXPATH], outSuffix[MAXPATH];
    memImage *outImage;
    int blur;
    sprintf(outSuffix, "%s%0", "b");
    makePath(outPath, outputDir, prefix, frameCounter,
        outSuffix);
    outImage = new memImage(imHeight, imWidth, bpp);
    BYTE red, green, blue;
    for (row = 1; row < imHeight; row++) {
        for (col = 1; col < imWidth; col++) {
            int bucket = 0;
            int redBucket = 0;
            int greenBucket = 0;
            int blueBucket = 0;
            for (blur = 0; blur < blurDepth; blur++) {
                switch (bpp) {
                    case 8:
                        bucket += images[blur]->getMPixel(col, row);
                        break;
                    case 24:
                        images[blur]->getMPixelRGB(col, row, &red, &green,
                            &blue);
                        redBucket += red;
                        greenBucket += green;
                        blueBucket += blue;
                        break;
                    default:
                        break;
                }
            } //end switch
            if (bpp == 8) {
                float avgBucket = bucket/blurDepth;
                outImage->setMPixel(col, row, (BYTE)(avgBucket +
                    0.5));
            }
            if (bpp == 24) {
                float avgRedBucket = redBucket/blurDepth;
                float avgGreenBucket = greenBucket/blurDepth;
                float avgBlueBucket = blueBucket/blurDepth;
                outImage->setMPixelRGB(col, row,
                    (BYTE)(avgRedBucket + 0.5),
                    (BYTE)(avgGreenBucket + 0.5),
                    (BYTE)(avgBlueBucket + 0.5));
            }
        } //end inner loop
    } //end outer loop
} //end frame loop
// Save the blurred image
sprintf(g_msgText, "Saving: %s", outPath);
statusPrint(g_msgText);
outImage->writeBMP(outPath);
delete outImage;
//end sequence loop
// Close the remaining images
for (i = 0; i < blurDepth; i++)
    delete images[i];
return NULL;
}
```

TREAPS

Listing One

```
/* An interface for ordered objects. A class that is to be used
 * as a key in a treap must implement this interface. */
public interface Ordered
{
    /* Compares two ordered objects. Returns the value 0 if this object equals
     * the argument; a value less than 0 if this object is less than argument;
     * and a value larger than 0 if this object is greater than the argument. */
    int compareTo(ordered anotherOrderedObject);
}
```

(continued on page 86)

Distributed Applications for a Connected World

You're developing and programming applications for the desktop, the networks and the Internet. Or, you're steering your enterprise into a connected future. You need a power tool that puts you in control of your destiny and at the center of today's complete set of development tools, middleware technology and enterprise services.

Welcome to Object World, the development tool for a connected world—your world.

The Conference

OT development, programming and business-driven solutions from the PC to the mainframe to the Internet.

The Exhibits

The next generation of tools and distributed applications —
Java, ActiveX, CORBA/IIOP,
DCOM and more.



Object World West '97

Conference: July 21-25, 1997 • Exposition: July 23-25, 1997 • Moscone Center • San Francisco, CA

Time-limited conference discounts! Go online to register now! www.ow.com

CONFERENCE HOTLINE For information on sessions and speakers, call 617-433-1830.

For information on exhibiting, call 617-433-1569.

(continued from page 84)

Listing Two

```
import order.*;
import java.util.Enumeration;

class Year implements Ordered
{
    int year;
    Year(int y) { year = y; }
    public int compareTo(Ordered y) {
        if (year > ((Year) y).year)
            return 1;
        else if (year < ((Year) y).year)
            return -1;
        else
            return 0;
    }
    public String toString() {
        return Integer.toString(year);
    }
}

class TreapDemo
{
    public static void main(String[] args) {
        Treap treap = new Treap();
        treap.put(new Year(1935), "Elvis Presley");
        treap.put(new Year(1926), "Chuck Berry");
        treap.put(new Year(1941), "Bob Dylan");
        treap.put(new Year(1936), "Roy Orbison");
        treap.put(new Year(1915), "Muddy Waters");
        Enumeration k = treap.keys();
        Enumeration e = treap.elements();
        while (k.hasMoreElements()) {
            System.out.print(k.nextElement() + " ");
            System.out.println(e.nextElement());
        }
    }
}
```

INTERNET PROGRAMMING

Listing One

```
/* Ping.cpp */

#define WIN32_EXTRA_LEAN
#include <windows.h>
#include <winsock.h>
#include <wsipx.h>
#include <conio.h>
```

```
#include "aes.h"

#define SERVER_SOCKET 0x3200

SOCKET g_skt;
BOOL g_serverUp;
AES *g_aesp;
AES::AESHandle g_hEvent;

void ServerThread(void *)
{
    if (!g_serverUp)
        return;
    fd_set rdSkt;
    FD_ZERO(&rdSkt);
    FD_SET(g_skt, &rdSkt);
    struct timeval tv = {5,0};
    int rval = select(0, &rdSkt, 0, 0, &tv);

    // schedule another pool thread to listen
    g_hEvent = g_aesp->ScheduleEvent(0, ServerThread, 0);

    // use this thread to process client request to completion
    if (rval < 0)
        printf("Winsock: select returned %d.\n", WSAGetLastError());
    else if (rval > 0)
    {
        SOCKADDR_IPX addr;
        int recvLen, addrLen = sizeof addr;
        char recvBuf[5];
        if ((recvLen = recvfrom(g_skt, recvBuf, sizeof recvBuf, 0,
            (struct sockaddr *)&addr, &addrLen)) == SOCKET_ERROR)
        {
            printf("Winsock: recvfrom returned %d.\n", WSAGetLastError());
            return;
        }
        char *resp = strcmp(recvBuf, "Ping") ? "101 Invalid Request." : "Pong";
        int respLen = strlen(resp) + 1;
        if (sendto(g_skt, resp, respLen, 0, (struct sockaddr *)&addr,
            sizeof addr) == SOCKET_ERROR)
            printf("Winsock: sendto returned %d.\n", WSAGetLastError());
    }
    return;
}

BOOL StartServer(WORD wSocket)
{
    int err;
    SOCKADDR_IPX addr;
    WSADATA wsaData;

    if (!(g_aesp = new AES))
    {
        printf("Unable to create AES object.\n");
        return FALSE;
    }
    if (err = WSASStartup(MAKEWORD(1,1), &wsaData))
```

NEW RELEASE! The Al Stevens Cram Course on C/C++

- A collection of 24 exciting developments in language technology

Al Stevens, a world renowned programming expert and contributing editor for *Dr. Dobb's Journal*, has created the CD-ROM to answer all your C/C++ programming questions – the *Al Stevens Cram Course on C/C++*.

This CD-ROM includes the complete text of three books written by Al Stevens, an interactive step-by-step tutorial with precise explanations, video clips of Al discussing important topics, the GNU Compiler Suite directly connected to the exercises, lots of usable source code, plus a memory feature that bookmarks your place for quick returns to prior sessions.

Use this CD-ROM on your programming projects to answer questions, work through actual examples, compile your example source code, and make sure it's correct right then and there.

No matter what your level of programming expertise, the *Al Stevens Cram Course on C/C++* will help you become a more proficient programmer.

Includes these Books by Al Stevens:

Welcome to Programming;
Al Stevens Teaches C;
Teach Yourself ...C++.

This multimedia

CD-ROM features:

- Three Complete Programming Books
- Interactive Tutorials
- Bookmark for Quick Returns
- Video Clips
- Print and Copy/Paste functions
- Built in Compiler

Designed for Win95 and NT (will work on 3.1)

Price: \$69.95

To Order Call:

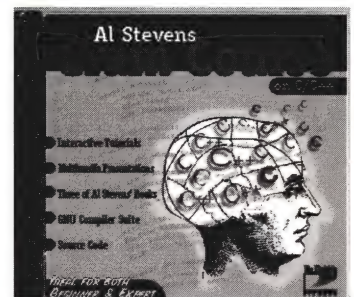
800-992-0549 (U.S. & Canada)
913-841-1631 (all other countries)

E-mail: orders@mfi.com

Fax Orders: 913-841-2624

Mail Orders:

Dr. Dobb's CD-ROM Library
1601 West 23rd Street, Ste. 200
Lawrence, KS 66046-2700 USA



See screen shots and a more detailed explanation on our Website:
www.ddj.com/cdrom


```

{
    printf("WSAStartup returned %d.\n", err);
    goto errout1;
}
if (LOBYTE(wsaData.wVersion) < 1 || HIBYTE(wsaData.wVersion) < 1)
{
    printf("Winsock version too low.\n");
    goto errout2;
}
if ((g_skt = socket(AF_IPX, SOCK_DGRAM, NSPROTO_IPX)) == INVALID_SOCKET)
{
    printf("Winsock: socket returned %d.\n", WSAGetLastError());
    goto errout2;
}
memset(&addr, 0, sizeof(SOCKADDR_IPX));
addr.sa_family = AF_IPX;
addr.sa_socket = htons(wSocket);
if (bind(g_skt, (struct sockaddr *) &addr, sizeof(SOCKADDR_IPX)) != 0)
{
    printf("Winsock: bind returned %d.\n", WSAGetLastError());
    goto errout3;
}
g_serverUp = TRUE;
if (!g_hEvent = g_aesp->ScheduleEvent(0, ServerThread, 0))
{
    printf("AES: Unable to schedule event.\n");
    g_serverUp = FALSE;
    goto errout3;
}
return TRUE;
errout3:
closesocket(g_skt);
errout2:
WSACleanup();
errout1:
delete g_aesp;
return FALSE;
}
void StopServer(void)
{
    g_serverUp = FALSE;
    g_aesp->CancelEvent(g_hEvent);
    delete g_aesp;
    closesocket(g_skt);
    WSACleanup();
    return;
}
void main(void)
{
    if (StartServer(SERVER_SOCKET))
    {
        printf("Server running, press any key to terminate...\n");
        getch();
        StopServer();
    }
    printf("Server terminated, press any key to exit...\n");
    getch();
    return;
}

```

Listing Two

```

/* AES.cpp */
#include "AES.h"

#include <process.h>
#include <limits.h>
#include <rtldbg.h>

unsigned __stdcall AES::PoolThread::Worker(void *ref)
{
    PoolThread& This = *(PoolThread *)ref;
    while (!This.dying)
    {
        Event *event = This.event;
        This.event = 0;
        if (!event)
            WaitForSingleObject(This.sem, INFINITE);
        else
        {
            event->Work();
            delete event;
        }
    }
    return 0;
}

AES::PoolThread::PoolThread(int stack /* = 0 */) : dying(FALSE),
sem(0), handle(0), event(0), InitState(0)
{
    if (!sem = CreateSemaphore(0, 1, 0, 0))
        !! (handle = (HANDLE)_beginthreadex(0, stack,
        Worker, 0, 0, 0));
    InitState = ERR_CANT_CREATE_HANDLE;
    return;
}

AES::PoolThread::~PoolThread(void)
{
    dying = TRUE;
    if (sem)
    {
        if (handle)
        {
            ReleaseSemaphore(sem, 1, 0);
            WaitForSingleObject(handle, INFINITE);
            CloseHandle(handle);
        }
        CloseHandle(sem);
    }
    return;
}

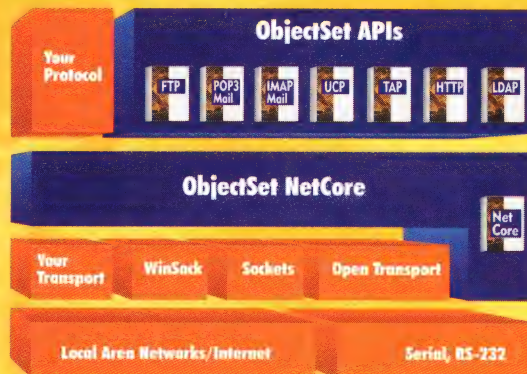
```

(continued on page 88)

Looking for a fast way to get your apps to communicate?

Develop your own protocols on top of NetCore. We'll handle all the transport, encoding / decoding, and memory / file management issues for you.

Our protocol APIs have easy functions that save you time. No experience required.



Our unique stream- and socket-oriented design is powerful, efficient, and easy to use.

Try this completely open C++ framework to handle all your protocol and transport needs.

ObjectSet NetCore

That's what the ObjectSet product line is all about. You'll be able to use our components to write software with powerful communication features on any platform. Regardless of transport or protocols. Save development time by using ObjectSet protocol APIs for e-mail, FTP, HTTP, and more. Or reap the benefits of open design by plugging your own protocols and transport layers right into the NetCore framework. ObjectSet products support standard TCP/IP stacks and serial lines. But if that's not enough, just integrate your own transport layer and keep using the same application and protocol code. ObjectSet is available on many platforms, and can be ported easily to many others. Of course, all ObjectSet products are royalty free.



If you want your software to communicate with the speed of your own ideas, try executing your ideas with ObjectSet.

ObjectSet Mail	ObjectSet IMAP	ObjectSet FTP	ObjectSet HTTP	Also available:
MIME, SMTP, and POP3 classes to send/receive Internet e-mail. Handles all MIME features, large attachments, and complex embedded bodyparts. Automatically encodes and decodes all attachments.	IMAP4, MIME, SMTP, and POP3 classes. Same as above but with full IMAP4 support.	C++ classes to add reliable and efficient file transfer capabilities to your own applications.	C++ classes for Web-enabling applications. Sends Requests and Posts to HTTP servers. Handles HTTP services. Decodes HTML content. Manipulates URLs.	LDAP, TAP, and UCP classes. For selected protocols we also have ActiveX controls, 16- and 32-bit DLLs, and attachable scripts for Macintosh.



Smartcode Software, Inc. Phone: (847) 945-3516 - Fax: (847) 945-6889
<http://www.smartcodesoft.com> - osmail@smartcodesoft.com
 In Europe: Phone: +33 4 67 59 30 40 - Fax: +33 4 67 59 30 48
<http://www.smartcode.fr> - osmail@smartcode.fr

CIRCLE NO. 76 ON READER SERVICE CARD

Communications?

Compression?

Lots of

New Stuff

From Greenleaf

Visit us today...

<http://www.gleaf.com/~gleaf>

NEW—ViewComm for Windows

Asynch communications debugger & protocol analyzer for Windows 95 and NT. See data and control signals on RS-232 links in real-time or review captured data. Flexible trigger system provides start-stop control of display or capture. Graphical display of modem & control signals. Data display in hex, octal, decimal, binary. ASCII & EBCDIC data. Character or hex displays. Searches. Many extras. ViewComm/DOS also available. Download TestDrive today:

<http://www.gleaf.com/~gleaf>

NEW—ArchiveLib 2.1

Greenleaf proprietary and PKZIP 2.0x compatible compression engines. Flexible archiving including compatibility with PKZIP. Archive to memory or file. PKWARE compatible encryption. Visual Basic, Pascal, Delphi, and C/C++ APIs. Supports Windows 95, NT, Windows 3.1x, OS/2 Warp, and 16- and 32-bit DOS.

Greenleaf Comm++ 3.0

Asynch communications the C++ way. Classes for file transfer, terminal emulation, screen drivers, modem and port control, flow control. Supports 16550, smart and non-smart multi-port boards. Supports Windows 95, NT, Windows 3.1x, OS/2 Warp, Novell NASi and 32-bit DOS.

No royalties. Lots of examples. Onscreen documentation. Library products come with free source code. 30-day no-questions money-back. Download free test-drive software.

☎ Call today for complete information, demo, or to order. MasterCard, VISA, AmEx and Novus; approved PO's

1-800-523-9830

Greenleaf Software, Inc.

16479 Dallas Parkway, Suite 570

Dallas, TX 75248

800.523.9830 972.248.2561

FAX 972.248.7830

BBS 972.250.3778 14:4/8/N/1

<http://www.gleaf.com/~gleaf>

email info@gleaf.com



(continued from page 87)

```
unsigned __stdcall AES::Monitor(void *ref)
{
    AES &This = *(AES*)ref;
    _CrtSetReportMode(_CRT_WARN, _CRTDBG_MODE_DEBUG);
    while (!This.dying)
    {
        int eventsWaiting = 0;
        ULONG now = ULONG(clock() / CLOCKS_PER_SEC);
        ULONG nextEventTime = ULONG_MAX; lastComplaintTime = 0;
        EnterCriticalSection(&This.critSec);
        for (EventList::iterator eli = This.eventList.begin();
            eli != This.eventList.end(); /**/)
        {
            for (int i = 0; i < This.size && This.pool[i].Busy(); i++)
            {
                // find next available worker
                if (i < This.size && (*eli)->When() <= now)
                {
                    Event *event = *eli;
                    eli = This.eventList.erase(eli);
                    This.pool[i].Alloc(event);
                }
            }
            else
            {
                if ((*eli)->When() <= now)
                    eventsWaiting++;
                else if ((*eli)->When() < nextEventTime)
                    nextEventTime = (*eli)->When();
                eli++;
            }
        }
        LeaveCriticalSection(&This.critSec);
        if (!eventsWaiting)
            WaitForSingleObject(This.sem, (nextEventTime - now) * 1000);
        else
        {
            if (lastComplaintTime + 3 < now)
            {
                _CrtDbgReport(_CRT_WARN, 0, 0, 0,
                    "Thread pool overflow. %d events delayed.", eventsWaiting);
                lastComplaintTime = now;
            }
            Sleep(100);
        }
    }
    return 0;
}

AES::AESHandle AES::ScheduleEvent(ULONG ulDelay,
    void (*pFunc)(void*), void *pData)
{
    Event *event;
    _ASSERT(pFunc);
    if (!event = new Event(ulDelay, pFunc, pData))
        return AESHandle(0);
    EnterCriticalSection(&critSec);
    eventList.push_front(event);
    LeaveCriticalSection(&critSec);
    ReleaseSemaphore(sem, 1, 0);
    return AESHandle(event);
}

BOOL AES::CancelEvent(AESHandle hEvent)
{
    Event *event = (Event *)hEvent;
    BOOL cancelled = FALSE;
    EnterCriticalSection(&critSec);
    for (EventList::iterator eli = eventList.begin();
        eli != eventList.end(); eli++)
        if (*eli == event)
        {
            eventList.erase(eli);
            delete event;
            cancelled = TRUE;
            break;
        }
    LeaveCriticalSection(&critSec);
    return cancelled;
}

AES::AES(int threads /* = 10 */ : dying(FALSE), sem(0),
    handle(0), size(threads), pool(0), initState(0))
{
    // Initialize event list critical section
    InitializeCriticalSection(&critSec);
    // Allocate thread pool
    if (!(pool = new PoolThread[size]))
    {
        initState = ERR_INSUFFICIENT_MEMORY;
        return;
    }
    // Check init state of threads in pool
    for (int i = 0; i < size; i++)
        if (pool[i].InitState)
        {
            initState = pool[i].InitState;
            return;
        }
    // Create monitor semaphore and thread
    if (!(sem = CreateSemaphore(0, 0, 1, 0)) ||
        !(handle = (HANDLE)_beginthreadex(0, 0, Monitor, this, 0, 0)))
        initState = ERR_CANT_CREATE_HANDLE;
    return;
}

AES::~AES(void)
{
    dying = TRUE;
    if (sem)
    {
        if (handle)
        {
            ReleaseSemaphore(sem, 1, 0);
            WaitForSingleObject(handle, INFINITE);
            CloseHandle(handle);
        }
        CloseHandle(sem);
    }
}
```



```

}
delete [] pool;
DeleteCriticalSection(&critSec);
return;
}

```

RSX

Listing One

```

// Set up output
output.FileName=c:\noises.wav
output.ChannelCount=2
output.SamplesPerSec=44100
output.BitsPerSample=16
output.TotalPlayTime=30.0f
output.PeripheralType=Headphones
environment.Reverb(0.0f, TRUE, 1.0f, 0.01f)
environment.SpeedOfSound(0.0f, 525.0f)

emitter n1=c:\windows\media\Robotz Windows Start.wav
emitter n2=c:\windows\media\Utopia Recycle.wav
emitter n3=c:\windows\media\Utopia Windows Start.wav
emitter n4=c:\windows\media\Utopia Windows Exit.wav

n1.Model( 0.0f, 1.0f, 30.0f, 1.0f, 30.0f, 1.0f)
n2.Model( 0.0f, 1.0f, 22.0f, 1.0f, 22.0f, 1.0f)
n3.Model(0.0f, 1.0f, 85.0f, 1.0f, 85.0f, 1.0f)
n4.Model(0.0f, 1.0f, 100.0f, 1.0f, 100.0f, 1.0f)

n1.ControlMedia(0.0f, RSX_PLAY, 0, 0.0f)
n2.ControlMedia(0.0f, RSX_PLAY, 0, 0.0f)
n3.ControlMedia(0.0f, RSX_PLAY, 0, 0.0f)
n4.ControlMedia(0.0f, RSX_PLAY, 0, 0.0f)

// Initial positions -----
n1.Position( 0.0f, 50.0f, 0.0f, 0.0f)
n2.Position( 0.0f, 100.0f, 0.0f, 0.0f)

n4.Position( 0.0f, 0.0f, 5.0f, 25.0f)
n4.Position( 3.0f, 150.0f, 5.0f, 25.0f)
n4.Position( 4.0f, 150.0f, 5.0f, -25.0f)
n4.Position( 7.0f, 0.0f, 5.0f, -25.0f)
n4.Position( 8.0f, 0.0f, 5.0f, 25.0f)
n4.Position(11.0f, 150.0f, 5.0f, 25.0f)
n4.Position(12.0f, 150.0f, 5.0f, -25.0f)
n4.Position(15.0f, 0.0f, 5.0f, -25.0f)
n4.Position(16.0f, 0.0f, 5.0f, 25.0f)
environment.Reverb(17.0f, TRUE, 1.0f, 0.1f)
n4.Position(19.0f, 150.0f, 5.0f, 25.0f)
n4.Position(20.0f, 150.0f, 5.0f, -25.0f)
n4.Position(23.0f, 0.0f, 5.0f, -25.0f)
n4.Position(24.0f, 0.0f, 5.0f, 25.0f)
n4.Position(27.0f, 150.0f, 5.0f, 25.0f)
n4.Position(28.0f, 150.0f, 5.0f, -25.0f)

n3.Position( 0.0f, 0.0f, 0.0f, 25.0f)
n3.Position( 5.0f, 150.0f, 0.0f, 25.0f)
n3.Position( 6.0f, 150.0f, 0.0f, -25.0f)
n3.Position(11.0f, 0.0f, 0.0f, -25.0f)

n3.Position(12.0f, 0.0f, 0.0f, 25.0f)
n3.Position(17.0f, 150.0f, 0.0f, 25.0f)
n3.Position(18.0f, 150.0f, 0.0f, -25.0f)

n3.Position(23.0f, 0.0f, 0.0f, -25.0f)
n3.Position(24.0f, 0.0f, 0.0f, 25.0f)
n3.Position(29.0f, 150.0f, 0.0f, 25.0f)

// Start-em up!-----
listener.Orientation(0.0f, 1.0f, 0.0f, 0.0f, 0.0f, 1.0f, 0.0f)

// Move the listener around
listener.Position(0.0f, 10.0f, 5.0f, 0.0f)
listener.Position(2.0f, 40.0f, 5.0f, 0.0f)

// Circle once
listener.Position(4.0f, 50.0f, 5.0f, -10.0f)
listener.Position(6.0f, 60.0f, 5.0f, 0.0f)
listener.Position(8.0f, 50.0f, 5.0f, 10.0f)
listener.Position(10.0f, 40.0f, 5.0f, 0.0f)

// Circle twice
listener.Position(12.0f, 50.0f, 5.0f, -10.0f)
listener.Position(14.0f, 60.0f, 5.0f, 0.0f)
listener.Position(16.0f, 50.0f, 5.0f, 10.0f)
listener.Position(18.0f, 40.0f, 5.0f, 0.0f)

// Circle thrice
listener.Position(20.0f, 50.0f, 5.0f, -10.0f)
listener.Position(22.0f, 60.0f, 5.0f, 0.0f)
listener.Position(24.0f, 50.0f, 5.0f, 10.0f)
listener.Position(26.0f, 40.0f, 5.0f, 0.0f)

listener.Position(28.0f, 90.0f, 0.0f, 0.0f, 0.0f)
listener.Orientation(27.0f, 1.0f, 0.0f, 0.0f, 0.0f, 1.0f, 0.0f)
listener.Orientation(28.0f, 0.0f, 0.0f, 1.0f, 0.0f, 1.0f, 0.0f)

```

Listing Two

```

MyEmitter1.Position(2.0f, 0.0f, 0.0f, 0.0f)
MyEmitter1.Position(5.0f, 4.0f, 4.0f, 4.0f)

```

(continued on page 90)

Work there, make waves. Work here, start tsunamis.

Imagine helping Fortune 1000 companies push past their own power curves? Imagine being turned to for support by the world's leading software engineers. That's life in our Enterprise Customer Unit. Got a passion for perfection? Superb decision making capabilities? A desire to show the biggest names in business what you're made of? No one can take you farther than Microsoft.

Application Development Consultants (Openings in Seattle, WA)

Help with deployments in large corporate accounts where Microsoft technology solutions are used. Conduct application design and code reviews, performance benchmarks, and provide porting/migration assistance, configuration management and general development consulting. Developer-level IBM or UNIX system background preferred. Strong business background in Fortune 500 and/or experience with systems technology consulting firm desired.

Competitive Technical Managers (Openings in Seattle, WA)

Drive the adoption of Microsoft products in enterprise accounts by overcoming technical objections based on comparison with major competitors. Extensive experience in Unix and mainframes, distributed communications architecture or Enterprise Consoles required. Seeking candidates with advanced degree in CS, or business with 8-10 years technical sales/development, support or consulting with large corporate customers. Extensive knowledge of Netscape or Oracle systems and knowledge of Microsoft products and technologies required. International experience highly desirable.

Developer Support Engineers (Openings in WA and NC)

Put your coding abilities and critical thinking to the test as you work with the latest technologies to provide solutions for other developers' toughest issues. Includes writing sample code, debugging customer code, and analyzing Microsoft source code. Must have solid coding experience in VB and/or C/C++ and the ability to grasp new technology quickly, excellent communication skills and a burning desire for customer satisfaction.

Technical Account Managers-Development Focus (Openings in WA, TX and NC)

Assist our Fortune 1000 corporate customers in the use and implementation of Microsoft solutions. Must have BS degree and 2+ years corporate MIS experience. Experience with networking architecture and added technical competence in one of the following areas: programming concepts (VB/C/C++); messaging architecture; or database architecture (SQL/Oracle). MCSE/MCSD certification desired.

SQL Course Developer/Trainer (Opening in WA)

Research, develop and deliver training on cutting edge SQL Server technologies. Combine your demonstrated SQL Server experience with solid project management, research and communication skills. Technical background must include experience with SQL Server, ODBC and transaction processing. Must have BS CS/EE and excellent written and verbal communication skills, including good presentation and facilitation skills. Must have the desire to facilitate technical development of others and be able to organize information for effective teaching.

OEM PM (Openings in WA)

Manage pre-release development between Original Equipment Manufacturers and Microsoft to achieve and implement design wins. Monitor some testing of OEM hardware configurations and Microsoft products. Qualifications should include a minimum of 2 years experience with computer hardware, desktop and corporate systems, development methods, and object management. Knowledge of OEM business practices and issues, hardware problems, and resolutions desired. Must be self managed and able to deal with people in time-demanding situations. A BA/BS degree in Computer Science or equivalent experience preferred. Ability to travel required.

Microsoft offers a competitive salary, excellent benefits, and substantial relocation packages. E-mail your resume in ASCII text format to: hireme@microsoft.com (indicate Dept. A28w9-0701 within the text of your resume) or fax to: (206) 704-4334 or mail to: Microsoft Corporation, Attn: Recruiting A28w9-0701, One Microsoft Way, STE 303, Redmond, WA 98052-8303. No phone calls please. We are an equal opportunity employer and support workforce diversity.

Microsoft

www.microsoft.com/jobs/

C++/MFC and Java Class Libraries

Stingray offers a complete family of object oriented C++/Java class libraries that speed your GUI application development!

Visual C++/MFC Class Libraries

★ Objective Grid 5.0

A full featured 100% MFC grid control - supports UNICODE, ODBC/DAO/OLE DB, find/replace, sorting, can be bound to any data source, and hundreds of other features!

★ Objective Toolkit 5.0

A "must have" set of over 60 MFC extensions including docking windows, smart edits, tabbed windows, tree, customizable toolbar and much more!

★ Objective Plug-in 1.0

Internet enable any MFC application by quickly converting it into a plug-in. Let our wizard generate a starter application for you - works with both IE and Navigator!

★ Objective Chart 1.0

The first 100% MFC, object-oriented charting/graphing class library. Objective Chart can be bound to Objective Grid and also supports a variety of dynamic/static 2d/3d graph types!

★ Objective Diagram 1.0

Add graphical diagram interfaces to your MFC applications quickly and easily with Objective Diagram! The Designer application helps you design custom graphical tool palettes palettesols.

Java Class Libraries

★ Objective Blend 1.1

An indispensable collection of over 15 AWT extensions that supercharge your Java applets. Includes: tree, list, tab control, masked edit, smart edits, progress and more!

★ Objective Grid/J 1.1

A 100% Java version of our popular C++ grid control. We also offer a Microsoft AFC compatible version!!

All products include FULL SOURCE CODE, 30-day money back guarantee, 60 days of free technical email support. Subscriptions and bundle/site discounts are available! All credit cards are accepted - ORDER TODAY!

For FREE DEMOS Surf to:

WWW.STINGSOFT.COM



"We add class to MFC/Java!"

Stingray Software, Inc.

1-800-924-4223

Voice: (919) 461-0672

Fax: (919) 461-9811

Email: sales@stingsoft.com

(continued from page 89)

Listing Three

```
// Output Settings
output.FileName=c:\mywave.wav
output.ChannelCount=2
output.SamplesPerSec=22050
output.BitsPerSample=16
output.TotalPlayTime=30.0f
output.PeripheralType=Headphones

// RSX environment
environment.Reverb(time, true/false, delay, intensity)
environment.SpeedOfSound(time, speed)

// Listener
listener.Position(time, x, y, z)
listener.Orientation(time, x, y, z, xUp, yUp, zUp)

// Emitters
emitter Emitter0=c:\sounds\bird.wav

Emitter0.Position(time, x, y, z)
Emitter0.Orientation(time, x, y, z)
Emitter0.Pitch(time, pitch)
Emitter0.Model(time, MaxFront, MinFront, MaxBack, MinBack, Intensity)
Emitter0.ControlMedia(time, action, loops, startPos)
Emitter0.SetMuteState(time, true/false)
Emitter0.SetMarkPosition(time, startPos, endpos)
```

TECATE

Listing One

```
clone moo WorldViewer
add moo {
  behavior {
    bootstrap (remoteHost shape) {
      # Alert a running session on remoteHost that we want to play
      if ($remoteHost != [gethost]) {
        send $remoteHost:[getself] enter "[gethost] ($shape)"
      } else {
        addstate shape $shape
      }
    }
    enter (host shape) {
      # Register the given host as a player, and let everyone else know
      set hostlist [getstate hosts]
      foreach i $hostlist {
        send [lindex $i 0]:[getself] newPlayer "$host ($shape)"
      }
      send [getsender] startUp "([lappend hostlist
        "[gethost] ([getstate shape])"])"
      send [getself] newPlayer "$host ($shape)"
    }
    newPlayer (host shape) {
      # Add the given host as a new player
      set hostlist [getstate hosts]
      addstate hosts [lappend hostlist "$host ($shape)"]
      puts "([gethost]) newPlayer: creating object $host"
      clone $host Entity
      add $host "appearance ($shape)"

      set position [get [getstate camera] state position]
      set orientation [get [getstate camera] state orientation]
      send $host:[getself] updateState "[gethost] [getstate room]
        ($position) ($orientation)"
    }
    startUp (hosts) {
      # Initialize play
      addstate hosts $hosts
      foreach i $hosts {
        set host [lindex $i 0]
        set shape [lindex $i 1]
        puts "([gethost]) startUp: creating object $host"
        clone $host Entity
        add $host "appearance ($shape)"
      }
      send [getself] broadcast {}
    }
    move {} {
      # Move self if flying
      inherited move {}
      send [getself] broadcast {}
    }
    requestUpdates {} {
      # Request an update from every other player
      foreach i [getstate hosts] {
        set host [lindex $i 0]
        send $host:[getself] updateMe
      }
    }
    updateMe {} {
      # Sender wishes to receive update of state
      set position [get [getstate camera] state position]
      set orientation [get [getstate camera] state orientation]
      send [getsender] updateState "[gethost] [getstate room]
        ($position) ($orientation)"
    }
    broadcast {} {
      # Broadcast current position and orientation to everyone else
      set position [get [getstate camera] state position]
      set orientation [get [getstate camera] state orientation]
      foreach i [getstate hosts] {
        set host [lindex $i 0]
        send $host:[getself] updateState "[gethost] [getstate room]
          ($position) ($orientation)"
      }
    }
  }
}
```



```

updateState (host room position orientation) {
# Update the position and orientation of the player on the given host
if ($room == [getstate room]) {
    set transform [mconcat [mtranslate $position] "$orientation"]
    add [getstate scene] "subobject ($host ($transform))"
} elseif ([lsearch [inquire [getstate scene]subobject] $host] != -1) {
    remove [getstate scene] "subobject ($host)"
}
}
}
}

```

Listing Two

```

clone Room Entity
add Room {
    state {
        scene ""
        window ""
        realized 0
    }
    behavior {
        init (scene window size color) {
            # Initialize the room
            addstate scene $scene
            addstate window $window
            addstate size $size
            add [getself] "appearance {
                shape (box [getstate size])
                diffusecolor ($color)
            }"
        }
        inLimits (x y z) {
            # Check if given point is inside or outside the room
            set width [lindex [getstate size] 0]
            set height [lindex [getstate size] 1]
            set depth [lindex [getstate size] 2]
            set min $width
            if ($height < $min) (set min $height)
            if ($depth < $min) (set min $depth)
            set pad [expr 0.05*$min]
            if ($pad < $x && $x < [expr $width - $pad]) {
                if ($pad < $y && $y < [expr $height - $pad]) {
                    if ($pad < $z && $z < [expr $depth - $pad]) {
                        return 1
                    }
                }
            }
            return 0
        }
        realize (target position up) {
            # Display the room and its contents
            addstate realized 1
            add [getstate scene] "subobject ([getself] ([midentity]))"
            set x [expr 0.5*[lindex [getstate size] 0]]
            set y [lindex [getstate size] 1]
            set z [expr 0.5*[lindex [getstate size] 2]]
            send [getstate window]-light(2) setView "{($x [expr 0.5*$y] 0.0)
                ($x $y $z) (0 1 0)}"
            send [getstate window]-light(3) setView "{($x [expr 0.5*$y] 0.0)
                ($x $y $z) (0 1 0)}"
            set camera [get [getstate window] state camera]
            send $camera setView "($target) ($position) ($up)"
            add [getstate window] "state (defaultPos ($position)
                defaultTar ($target))"
            send [getstate window] setRoom "[getself]"
            send [getself] callback realize
        }
        unrealize () {
            # Disable the room's display
            addstate realized 0
            remove [getstate scene] "subobject ([getself])"
            send [getself] callback unrealize
        }
    }
}

```

Listing Three

```

clone Bridge Entity
add Bridge {
    state {
        portals {}
    }
    behavior {
        init (color room0 target0 position0 up0 room1 target1 position1 up1) {
            clone [getself]-0 Portal
            send [getself]-0 init "[getself] $room0 ($target0) ($position0)
                ($up0) ($color)"
            clone [getself]-1 Portal
            send [getself]-1 init "[getself] $room1 ($target1) ($position1)
                ($up1) ($color)"
            addstate portals "[getself]-0 [getself]-1"
            add [getself] "subobject ([getself]-0 [getself]-1)"
        }
        crossBridge (portal) {
            set i [lsearch [getstate portals] $portal]
            if ($i == -1) {
                puts "([getself]:oppositePortal) Unrecognized portal '$portal'"
            } else {
                set i [expr 1 - $i]
                set portal [lindex [getstate portals] $i]
                set position [get $portal state position]
                set target [get $portal state normal]
                set position [vadd $position [vscale 1.1 $target]]
                set target [vadd $position $target]
                send [get $portal state room] realize"($target)($position) (0 1 0)"
            }
        }
    }
}

```

DDJ

The hottest news in java...

is how easy it is to learn!



Introduction to
Programming Java Applets \$49.95



Java Programming & Core
Class Libraries \$49.95



Hands On Java
Programming, Using
Microsoft J++ \$39.95



Java Security
Managing the Risks \$59.95



Hands On Visual Cafe \$49.95



Inside JavaBeans \$59.95

If you're serious about staying up-to-date with the skills that today's marketplace demands, knowing the Java language is a must.

And there's no faster or easier way to master Java™ than with the MindQ® Java Series. This comprehensive set of tutorials on CD-ROM was developed by Java professionals and is the most effective approach to learning Java available today.

What makes the MindQ Java Series unique is that it combines hands-on practice and interactive multimedia to create a powerful, next generation learning experience.

Each CD-ROM contains animated code examples, step-by-step explanations and narrations that reinforce learning. And you'll learn through real world Java sample projects with complete source code listings.

Easy to understand and self-paced to suit your learning style, the MindQ Java Series is praised by Java pros and technology critics alike...

"A pleasant exception, surpassing PC Week Labs' expectation for CD-ROM courseware..." **PCWEEK**

So don't wait to get Java-literate and add to your arsenal of programming and web developing skills. Order the Java Series from MindQ today and you'll be on your way to becoming a more valuable technology professional.*



Visit our website for a demo of any MindQ title!

For Win '95 and NT

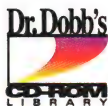
©1997 MindQ Publishing, Inc. All rights reserved. MindQ is a registered trademark of MindQ Publishing, Inc. Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries. Java is a trademark of Sun Microsystems, Inc. All other trademarks are property of their respective holders.

http://www.mindq.com
1-800-646-3008
*30 Day
Money Back
Guarantee

CIRCLE NO. 354 ON READER SERVICE CARD

Dr. Dobb's

Distributed Objects Development CD-ROM



The latest in distributed
object technology, including languages,
libraries, and environments



Programming distributed systems using traditional tools is both complex and error prone. That's why the editors of *Dr. Dobb's* assembled this collection. Using this CD-ROM will enable you to build reliable distributed, object-oriented applications. And since the Internet and World Wide Web are rapidly increasing the demand for distributed applications, these tools and utilities are at the very heart of this technology.

This CD-ROM contains over 100 megabytes, featuring:

- **Arjuna 3.3.2**—A robust O-O tool for building fault-tolerant distributed applications out of C++ programs
- **Shadows**—Creates World Wide Web Arjuna applications
- **Interlanguage Unification System (ILU)**—A powerful programming model for distributing applications via an Interface Description
- **Obliq**—An O-O scripting language supporting distributed computation
- **Visual Obliq**—The World Wide Web version of Obliq
- **Phantom**—A stand-alone, interpreted language for developing large-scale, interactive, distributed applications
- **Distributed Python**—A portable, interpreted, O-O language which is extendable



ONLY
\$39.95

For a detailed
description, please
visit our web site:
www.ddj.com/cdrom/

Order your
copy today!

FREE BONUS!

Get a FREE *Dr. Dobb's Sourcebook* on
Distributed Objects with
your CD-ROM purchase

CALL
TOLL FREE
800-500-6841

INT'L Orders:

Use mail, fax, e-mail, or call
913-841-1631

FAX: 913-841-2624

E-Mail Orders:

orders@mfi.com

Mail Orders:

Dr. Dobb's CD-ROM Library
1601 West 23rd St., Ste. 200
Lawrence, KS 66046-2703 USA

YES
Send me the *Dr. Dobb's Distributed Objects CD-ROM* immediately. I will also receive the
Dr. Dobb's Sourcebook, Distributed Objects as a bonus. I understand that I have a full money-back guarantee.

Distributed Objects Development CD-ROM.....\$39.95

Please add sales tax in the following states: CA (8.5%), GA (6%), IL (6.25%), KS (6.9%), NY (8.25%),
TX (8.25%), Canada (7% GST).

Shipping (\$2.00 for each CD-ROM shipped in USA/Canada, \$10.00 for Airmail to all other countries.)

TOTAL AMOUNT.....

Name

Address

City/State/Zip

Country

Phone

E-mail

Here's how I'm paying (circle one): VISA - MC - AMEX - Check enclosed

Credit card number (include all digits)

Expiration Date

Signature required

Please allow 2 weeks for delivery in the U.S./Canada; 4 weeks for all other countries. Checks/money orders must
be drawn on a U.S. bank in U.S. funds only.

DJ 97

Java Bashing, Books, and Beans

Michael Swaine

It is an unquestioned article of computer faith that James Gosling originally named the Java language "Oak" in honor of a tree outside his office window. According to this official Sun story, the name Java only emerged after a long trademark search.

I wonder.

I mean, think about it: We're talking about James Gosling, one of the most caffeinated programmers on the planet. Let me quote from the definition of "tense" in *The Hacker's Dictionary* (Guy L. Steele et al., Harper & Row, 1983):

This routine is so tense it will bring tears to your eyes. Much thanks to Craig Everhart and James Gosling for inspiring this hack attack.

And from the definition of "hack attack," so there can be no doubt about the state of wiredness being induced here:

I've been up for thirty hours; I had a hack attack and finished off that new feature I thought would take two weeks to program.

You get the picture, right? Since his MIT days, Gosling has been staying up late, staring bleary-eyed at a computer monitor, writing tense code that inspires hack attacks. And we all know what that means. Coffee. Possibly cappuccino.

Second point: Consider the picture we now have of Gosling, the bleary-eyed, caffeine-wired hacker, cranking out code so tense it brings tears to your eyes. I ask you, is this a person who stares out his office window? Is this a person who knows that he has an office window? Or is this a guy who invents a language and names it after his favorite drug? I think all this Oak business is just an urban legend.

Let It Be C

My recent rehashing of Ward Mullins' ten criticisms of Java inspired some reader mail.

Michael, DDJ's editor-at-large, can be contacted at mswaine@cruzio.com.

Bill Marsh thinks that only numbers one through four of Mullins' list of criticisms are valid complaints about the language, the other six being complaints about the atmosphere surrounding it. "I was surprised," Bill says, "that there was no mention of the ease of network programming with Java. This makes the language worthwhile by itself."

Indeed, Java benefits from having come into existence after the importance of being totally wired (Netsavvy, that is) had been made manifest. It was defined from the start (or from 1.0 anyway) to provide a lot of Net capabilities, with classes for creating and manipulating data packets, sockets, and URLs. Although Sun's claims about Java being easy to program in are exaggerated, it really is surprisingly easy to put together Internet applications and applets with Java. Right on, Bill.

Edward Diener thinks that all this fussing over Java's faults and virtues sounds a lot like certain articles that used to run in *Computer Language* magazine. Ed was apparently not a fan of *CL*. "Each issue seemed to feature some ludicrous debate about the best computer language in which to write a program. After a few issues of this, I ended my subscription. I have also done my best to avoid all debates about which language is better than which language. After all, could you imagine arguing with an Italian that Russian is better than his language? Or vice versa? Absurd!"

Actually, I can easily imagine an Italian and a Russian arguing over the relative virtues of their languages, although not as easily as I can imagine a Frenchman and a German doing so. Linguistic ethnocentrism is, if anything, more virulent for natural than for artificial languages. There's an interesting online publication out of Munich (<http://www.heise.de/tp/tpfhome.htm>) where you can read an article about how English is doomed to extinction because of the Web. The article, I need hardly mention, is written in English. But I digress.

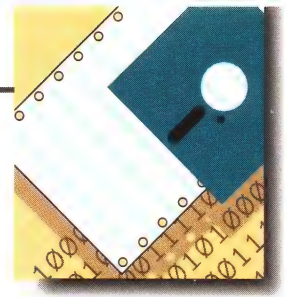
Say on, Ed Diener, say on.

"My biggest gripe regarding Java," Ed continues, "is that every mention of it comes with a corresponding putdown of C++. Why? Unless I am missing something deep, Java is about 85% C++ give or take 10%. If you like Java, enjoy yourself. I like C++. Maybe I'll learn Italian some day. Maybe you'll learn Russian. Who knows?"

Good call, that, about everybody gratuitously bashing C++ as they rush to deify Java. I may not be the best person to respond, since I never became a C++ fan, no doubt because I never got very good at programming in it. And Java is certainly highly similar to C++ in syntax, making my enthusiastic embracing of Java look a little silly. But I think the point Ed is missing is that the Java hoopla is not about Java the language, but about Java the phenomenon.

Java the phenomenon encompasses JavaOS the operating system, JavaBeans the software component architecture, Java microprocessors, and a plethora of APIs, as well as Java the language. Java the phenomenon is spreading like mold on the surface of a misplaced cup of coffee. Books on Java are popping up as fast as web books last year or Internet books the year before. Companies doing embedded applications, smartcards, and thin clients are committing their fortunes to Java-based technologies that are today more promise than delivery. Apple has latched onto Java as though it will be the salvation of the company. It's hard to understand why, because there's not much here that we haven't seen before. "Write once, run everywhere" sounds great, but it rests on an object model derived from C++ and a bytecode approach last seen in the UCSD p-system back in the '80s.

Sun is pushing Java as the answer to security on the Net, and there are several security measures built into Java and the applet model, but nothing that couldn't be implemented without Java.



JavaBeans provides an interesting model for creating component software, an exciting possibility, but so, in different ways, do ActiveX and OpenDoc.

The explanation for Java the phenomenon seems to have something to do with the right pieces being put together in the right way at the right time and promoted in the right way. "Write once, run anywhere" not only sounds good, it sounds like an answer to the Wintel monopoly. And Sun is brilliantly managing to associate the words "Java" and "security" in the minds of people who don't know what Java is or what the real Internet-security issues are.

Those people matter, the people who know frighteningly little about computers but control departmental budgets. They tend to latch onto a magic phrase that they think will keep them out of trouble. In the 1980s, they asked "Is it IBM compatible?" In the 1990s, they asked, "Does it run Windows?" And I can almost hear them starting to ask, without having a clue what it means, "Was it written in Java?"

I suspect that Ed will learn Java, but who knows? I know this: There is zero probability that I'm going to learn Russian.

William Beebe connects my Java jabs with the musical parodies I quoted from

Don Slepian and comes up with a little ditty of his own:

When my project's hit a brick wall
K&R, they come to me
Speaking words of wisdom
Code in C

Code in C, Code in C
Code in C, yeah, Code in C
Java's not the answer
Code in C

"Since programming and music are supposedly intertwined within the psyche of the programmer," William says, attempting to palm off one of the ripest chestnuts of computer lore, "maybe you should have a musical language parody contest."

Not a bad idea, William. See the last page of this issue.

Bashing Books

In trying to stay on top of the Java phenomenon, I read mailing lists, online pubs, magazines, and books, as well as my e-mail. The books almost invariably disappoint. The current crop, in fact every wave of them so far, seem more than usually rushed. Some semirandom examples:

Laura Lemay has become a technical-book publishing empire unto herself. Her name may not be as big a draw as the

ubiquitous Dummy, but it's creeping up there. One such entry is Lemay's *Teach Yourself Java for Macintosh in 21 Days* (Laura Lemay and Charles L. Perkins, with Timothy Webster, Hayden Books, 1996). Like all her books, it is thoughtfully organized and well written (although, in this case, not written entirely or even mainly by Lemay), but it is poorly edited.

There are syntactic errors in the code, errors in tables, and errors in the text, including mistakes that seriously affect the usefulness of the book. And there's no excuse for publishing code with syntactic errors. Zero syntax errors is easily achievable, so anything less is unacceptable.

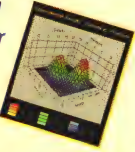
The Lemay book is not alone. In fact, despite its errors, I think it's one of the best Java books out there.

Freddy Hill writes, "An acquaintance of mine, Lewis, had purchased Norton's book on Java, and the book was filled with typographical errors in the code. Since Lewis was a novice, he could not find the errors; thus, he halted his attempts to learn Java."

I haven't read the Norton book, but Freddy's tale of woe doesn't surprise me. *Core Java*, Second Edition, by Gary Cornell and Cay S. Horstmann, SunSoft Press/Prentice Hall, 1997) is another highly regarded Java book, and deservedly so.

More Power More Freedom

Olectra Chart combines high performance graphing power with the freedom to easily create and customize virtually any type of chart.



Olectra Chart New Version 5.0

Advanced Controls Outperform the Competition

- High performance real-time charts
- Built-in zooming, scaling and rotation
- Annotate anywhere with unlimited labels
- Dynamic time and date axis on any plot
- Object-based C API, MFC and OWL wrappers
- 16 and 32-bit OCLs and DLLs
- Now with data binding!

Olectra Chart is 4X faster than First Impression and Graphics Server SDK in standard mode redraw speed.*

"...blazingly fast." Windows Tech Journal

Java Developers
Class Chart coming soon!

Motif Developers
Check out XRT products
at www.klg.com

KL GROUP
The Leader in GUI Components

CALL NOW TO ORDER! \$349*

1-800-709-2947

Download free trial version! <http://www.klg.com>

Money Back Satisfaction Guarantee!

KL Group Inc. Toll-Free: 1-800-709-2947 Tel: 416-594-1026 Fax: 416-594-1919
 KL Group Europe B.V. Tel: +31 (0)20 679 95 03 Fax: +31 (0)20 675 78 44
 info@klg.com www.klg.com

* Prices in US dollars. Prices may be higher outside North America. XRT, iClass and Olectra Chart are trademarks of KL Group Inc. All other product names are trademarks or registered trademarks of their respective owners. * Visit our Web site for more benchmark results. 007

CIRCLE NO. 288 ON READER SERVICE CARD

PCYACC® Version 7.0

PROFESSIONAL LANGUAGE DEVELOPMENT TOOLKIT

Includes "Drop In" Language Engines for SQL, dBASE, POSTSCRIPT, HYPERTALK, SMALLTALK-80, C++, C, PASCAL, PROLOG, FORTRAN, COBOL, BASIC, SGML, ASN, RPG, REXX, PL1, SNA, RTF, VISUAL BASIC, SQL2, DB2, VHDL, HTML, VMRL, JAVA, ODMG-ODL/OQL, SQL3, MODULA-3, DELPHI, VBS, and ADA.

PCYACC Version 7.0 is a complete Language Development Environment that generates C, C++, Java, Delphi, and VBS source code from input Language Description Grammars for building Assemblers, Compilers, Interpreters, Browsers, Page Description Languages, Language Translators, Syntax Directed Editors, Language Validators, Natural Language Processors, Expert System Shells, and Query languages.

- Portable Object Oriented Classes for C++ and JAVA - Error, Symbol Table, Syntax Tree, Yacc, and Lex.
- Debugging tools include runtime Visual Parser Debugging, Abstract Syntax Tree generation, and Cross Referencing.

CodeCheck® Version 7.0

SOURCE CODE ANALYST

Includes "Drop-In" Rules for Compliance analysis, Adherence to specifications, Measures of complexity, Silent error detection, Code maintainability, and Portability.

CodeCheck Version 7.0 is a programmable tool for managing all C and C++ source code on a file or project basis. CodeCheck is input compatible with all variants of K&R, ANSI C and C++ (Microsoft, Metaware, Borland, Intel, Vax/Vms, HP/Apollo, Microtec, Watcom, Symantec, MPW, CodeWarrior, AT&T, and GNU). CodeCheck is designed to solve all of your Portability, Maintainability, Complexity, Reusability, Quality Assurance, Style Analysis, Library/Class Management, Code Review, Software Metric, Standards Adherence, and Corporate Compliance Problems.

- Portability - CodeCheck identifies code that will not port between MSDOS, OS/2, Unix, VMS, WIN-95/NT, Macintosh and porting to 64 bit machines.
- Compliance - CodeCheck allows your corporate coding and project specification standards to be completely automated for compliance validation.

CodeCheck includes pre-written expert system Rule Files that can be applied to any C or C++ project for: POSIX, SVID, X3J11, Ellementel, and ANSI C/C++ Source Code Conformance Validation.

30 day Money back guarantee! Free AIR Shipping anywhere in the world!

	CodeCheck	PCYACC
DOS/WIN16	\$495	\$495
MAC	\$495	\$495
OS/2	\$995	\$995
WIN32 95/NT	\$995	\$995
UNIX/VMS	\$1995	\$1995



ABRAXAS
Software, Inc.

5530 SW Kelly Ave., Portland, OR 97201 USA
 TEL (503) 244-5253 • FAX (503) 244-8375
 E-mail: sales@abxsoft.com

www.abxsoft.com

CIRCLE NO. 124 ON READER SERVICE CARD

The authors know their stuff and share a lot of practical advice that could only come from being down in the trenches. They also speak honestly and thoughtfully about what they like and don't like about Java.

But the book is, again, not well edited. A good editor could have helped these guys with the organization of their material, and alerted them when they were repeating themselves unnecessarily.

A minor gripe: *Core Java* should really be called *Core Java for Windows 95 Programmers*. The examples, tips, and insights are nearly all Win95 specific. There are some allusions to Solaris and a fleeting acknowledgment that the authors have heard of something called a Macintosh. But this is really a Win95 book, a limitation nowhere indicated on the cover. There is nothing unusual in this, unfortunately. Computer-book publishers are getting more casual than ever about acknowledging such limitations. But in the platform-transcending atmosphere of the late 1990s, maybe we shouldn't let them get away with it.

Bean Book

Book authors and publishers have an insurmountable problem in a fast-moving area like Java programming. There's really no way they can keep up, and even the standard references go quickly obsolete. Version 1.1 changes so many things that those just getting started with Java now are perhaps lucky that they can just jump in at the 1.1 level. A warning: It's really important to publishers to have the latest version number on the cover, even if the author has only briefly seen a beta. Compare the copyright date with the release date of the new version before you accept the claims on the cover.

One Java book in which I haven't yet found any typos is Michael Morrison's *Presenting JavaBeans* (sams.net Publishing, 1997). This may just be because I've been reading so many books lately that I'm getting bleary-eyed (but not in the tense Gosling way; this is more of a loose goose blear).

JavaBeans is, of course, Sun's API for developing component software. It includes the needed interfaces and classes for creating beans, as these software components are called. *Presenting JavaBeans* does a good job of presenting JavaBeans, and includes four sample beans on the accompanying CD-ROM, including beans for an LED display and a simple audio player.

I like Morrison's organization. He starts off with 40 pages of introduction, in which he lays out the basic concepts of component software and the various facilities provided by the JavaBeans API, explaining what they're for and why you might

want to use them. Then he jumps right into the API, detailing its features so that you know what you've got to work with, and then, around page 125, he gets to some examples of bean construction.

Mac developers should note that the BDK, or Bean Developer's Kit, is not available for the Mac at the time I am writing. Versions for Windows 95 and NT and Solaris are included on *Presenting JavaBeans* accompanying CD, but you'll have to

JavaBeans provides an interesting model for creating component software

download the Mac version when it's available from Sun's site; it's free, though. And sams.net *does* acknowledge this on the cover of the book.

Seems like everybody is developing beans now, but that's probably an illusion of the SunSoft reality repurposing machine. Still you wonder when a company you thought was dead resurfaces as a bean factory (<http://www.taligent.com/>).

Morrison calls JavaBeans "possibly the most exciting and promising software component technology available." I had to check the publication date on this book myself; the "available" qualifier sounded just too appropriate. But no, Morrison delivered his manuscript before Apple announced that it was pulling out of OpenDoc development. Just prescient, I guess.

Because a reasonable argument can be given that OpenDoc was a particularly exciting and promising software component technology, albeit a component technology of a different sort.

The OpenDoc idea is, or was, to give components to mere users and let them configure their own integrated software. Sort of like UNIX pipes, only not really. More like audio components. I speak as though that option is now closed, but in fact OpenDoc is not dead. It's just been pulled off life support.

Apple's position, stated at one of this year's cutbacks, is that it's moving its com-

ponent technology resources from OpenDoc to Java, whatever that means, and that the releases of OpenDoc and CyberDog that ship with Mac OS 8 (formerly known as 7.something) will be the last major update to Mac OpenDoc. There will be no OpenStep API for OpenDoc. CI Labs has been disbanded and its assets returned to its sponsor companies, principally Apple and IBM. IBM may still be planning to support the Windows version of OpenDoc.

Black on Black

One phrase I like to keep in mind when thinking about component software is "how to hook a customer directly to a product." It's not terribly catchy or original or surprising, but it reminds me that there is a constant battle to eliminate the middleman, and that it can always be done.

The phrase is from Roger Black's *Web Sites that Work* (Roger Black and Sean Elder, Adobe Press, 1997). Black is a famous designer, mostly of magazines. I'd say more, but Black says it far better in the book than I can here. There are those who think that Black is just a bit, shall we say, full of himself. In this book, he gracefully acknowledges his coauthors and removes himself from the book in the introduction by the technique of banning the first person from the book. Or that's the theory, anyway. The third-personing of the author actually has a very different effect. Roger Black is one of the themes of the book. There's a chapter of Black's from-the-hip prognostications about the Web, a chapter of images from his past work, quotes about Black from the editors of *Wired*, and so on.

Nevertheless, the book is a real boon to anyone trying to make sense of web-page design. It's full of firm dos and don'ts, like "Don't repurpose," "Make everything as big as possible," "Put content on every page," and "The first color is white; the second color is black; the third color is red."

When you're dealing with design issues but aren't a professionally trained designer, what you want most is strong clear assertions that cover all the decisions facing you and eliminate the ambiguities and grey areas. And you don't care whether the assertions are the best possible views on the issues; you just want them to be not wrong.

That's Black.

By the way, regarding that urban legend of Java's naming, I have heard another theory of how the language got its moniker. This one says that Java is an acronym for "Just Another Virtual Architecture." That can't be true, can it?

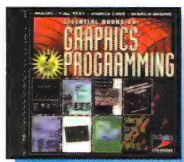
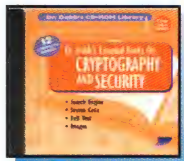
DDJ

Dr. Dobb's CD-ROM Library

The Software Professional's Essential Resource

Visit our website! www.ddj.com/cdrom/

The *Dr. Dobb's CD-ROM Library* is an extensive collection of powerful tools and resources, essential to every programmer's productivity and success. The library focuses on critical aspects of programming ranging from an expansive back-issue archive of *Dr. Dobb's Journal* to essential book selections on *Algorithms and Data Structures* and also *Graphics Programming*. And coming from *Dr. Dobb's*, you know that each CD-ROM is chock-full of source code. Take a moment to browse through our library. We guarantee your satisfaction.



Dr. Dobb's/CD Release 5 CD-ROM

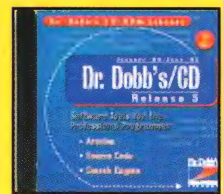
Dr. Dobb's/CD Release 5 provides you with immediate and complete access to nine and a half years of *Dr. Dobb's Journal*. Every Algorithm Alley, every Swaine's Flame, every article, and every bit of source code from January 1988 through June 1997 is included on this CD-ROM. In addition, *Dr. Dobb's/CD Release 5* contains all issues of *Dr. Dobb's Sourcebook*. The magazine devoted to spotting and analyzing emerging trends in software technology.

Bringing together a decade's worth of perspectives and expertise from the most respected programmers in the industry, *Dr. Dobb's CD/Release 5* offers both longtime and first-time readers an

unparalleled programming resource. Use its powerful search engine to find what you are looking for and then print it, copy/paste it, or directly export it as a file.

No more sifting through dozens of back issues and manually keying source code into your current programming project. With *Dr. Dobb's/CD Release 5*, you have it all when, where, and how you want it!

Release 5 includes these must-have features: super-fast full text, boolean, search engine, indexes for both articles and authors, copy/paste/print functionality, save searches for quick access later, annotate and bookmark for your future reference needs.



Price: \$49.95

FAST Search Engine

\$29.95 upgrade available to prior owners!

To upgrade call:
800-822-1162

Introducing Dr. Dobb's Essential Books on Cryptography And Security CD-ROM

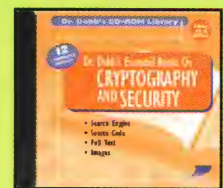
The editors of *Dr. Dobb's Journal* have selected the most important books on encryption technology and have compiled it into a resource every programmer must have. This CD-ROM presents both the theory and practice of network security implemented in C, BASIC, and other familiar programming languages.

Special Features include: Full-Text Search Engine, Complete text of all books, Hyperlinks across all books, Indexes for all books, Technical bulletins, Security briefs, and Cryptographic FAQs from RSA Data Security.

The *Essential Books on Cryptography and Security CD-ROM* provides detailed documentation and analysis of:

- Classical and Modern Encryption Algorithms
- Private-key and Public-key cryptography
- The Data Encryption Standard
- Signature schemes
- Block Ciphers and Stream Ciphers
- Plus Much, Much, More!

This CD-ROM will be shipping in July 1997. Reserve your copy now!



Price: \$99.95

New Releases!

ORDER TODAY!

U.S. & Canada:
800-992-0549

All Other Countries:
913-841-1631

E-mail:
orders@mfi.com

Fax Orders:
913-841-2624

Mail Orders

Dr. Dobb's CD-ROM Library
1601 West 23rd St., Ste. 200
Lawrence, KS 66046-2703 USA

Go to our website for more products from
Dr. Dobb's CD-ROM Library: www.ddj.com/cdrom/

Dr. Dobb's
CD-ROM
LIBRARY

Buy 2 CD's and get 25% off the 3rd!

Essential Books on File Formats CD-ROM

6 Essential Books selected by the Dr. Dobb's

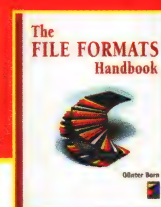
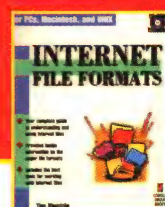
Look no further! The *Essential Books on File Formats CD-ROM* contains the complete text from six books, which will provide you with the most comprehensive and detailed information on all the important and popular file formats in use today. Selected by the editors of *Dr. Dobb's*, this CD-ROM contains invaluable information on file formats used for graphics, multimedia, sound, databases, spreadsheets, Windows, the Internet, and much more!

No matter what your programming focus, the *Essential Books on File Formats CD-ROM* discloses the secrets and insider knowledge you need to make your programming instantly compatible with all the major applications out there right from the start. Plus, the CD-ROM's powerful full-text search engine and hyperlink capabilities allow you to search quickly and easily across all the books to link directly to the information you need.



Price \$69.95

FAST Search Engine



Essential Books on Algorithms and Data Structures CD-ROM

8 Essential B

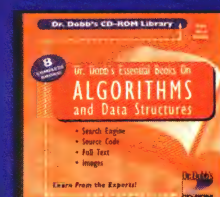
Have you ever been stumped when creating an algorithm or data structure? Then the *Essential Books on Algorithms and Data Structures CD-ROM* belongs on your desktop!

These eight books, selected by the *Dr. Dobb's* editors, are the most important books ever written on the subject and provide you, the software professional, with the details you need to succeed. Plus, this CD-ROM features a full-text search engine, giving you instant access to the entire contents of all eight books.

Learn from the experts. Set up your algorithms the right way, the first time — before you ever write that first line of code!

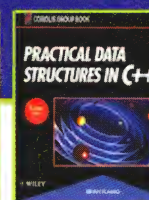
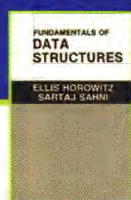
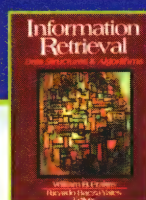
Your CD-ROM includes these must-have features:

- A Powerful Full-Text Search Engine
- Full Windows Functionality
- Hyperlinked TOCs and Indexes for Each Book
- Create Your Own Bookmarks and Annotations
- Plus Much More!



Price \$59.95

FAST Search Engine

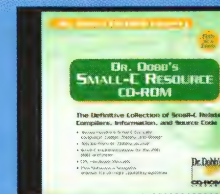


Dr. Dobb's Small-C Resource CD-ROM

The *Small-C Resource CD-ROM* contains the definitive collection of compilers, information, and source code for the Small-C compiler. Packed with books, tools, utilities, and Small-C implementations, this CD-ROM includes James Hendrix's out-of-print book, *A Small-C Compiler: Language, Usage, Theory, and Design* and several relevant

articles from *Dr. Dobb's Journal* written by C experts such as Allen Holub.

Plus all content is hyperlinked in HTML format, which makes viewing easy and convenient. And for those who don't have an HTML browser, included is Netscape's Navigator browser supporting all major operating systems.



Price \$49.95

ORDER TODAY!

U.S. & Canada:
800-992-0549

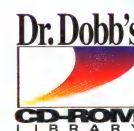
All Other Countries:
913-841-1631

E-mail:
orders@mfi.com

Fax Orders:
913-841-2624

Mail Orders

Dr. Dobb's CD-ROM Library
1601 West 23rd St., Ste. 200
Lawrence, KS 66046-2703 USA



Go to our website for more products from Dr. Dobb's CD-ROM Library: www.ddj.com/cdrom/

Netscape - [Raima Corp. - High Performance, Cross-Platform Database Development Tool]

File Edit View Go Bookmarks Options Directory Window Help

Go to: <http://www.raima.com>



If it's fast – it's RAIMA™

High Performance Database Technology

This just in...

-  **6/10/97** - Pure Atria's ClearCase and Wang's (now Kodak's) OPEN/image server-based software are recognized as leading Raima-based applications.
-  **5/28/97** - Velocis Database Server now bundled with Berkley Software Design, Inc.'s BSD/OS 3.0. Velocis and BSDI's Internet Suite provide a powerful Web solution.
-  **5/2/97** - Free downloads and try-and-buy software available from this Web site. Developers, check out this database technology today!
-  **4/28/97** - Vista Development, Raima's consulting subsidiary, adds Internet/Intranet solutions to its client/server database development services.
-  **4/6/97** - Raima's Report Writer includes a low-level C++ API for complete developer control when embedding Report Writer in applications.

About Raima



We pioneered the fast, low-level database engine, and offer the best database technology for application development.

Products & Services



High-performance client/server DBMS, real-time embedded database engine, reporting tools, Web database technology, and consulting services.

Support



World-class support with interactive product support forums, patch information, and pre-releases for our valued customers.

News & Publications



Developments in technology, the latest Raima-based applications, and news on upcoming conferences, user groups and events.

This site features dynamically generated content from Raima's Velocis Database Server (feel free to view the [SQL DDL schema](#) that we used). Velocis is accessed through our Velocis Web Server Gateway and Allaire's Cold Fusion.

Raima Corporation, 4800 Columbia Center, 701 Fifth Avenue, Seattle, WA 98104 USA
Inside the U.S.: 1-800-275-4724, Outside the U.S.: 1-206-515-9477, Fax: 1-206-748-5200

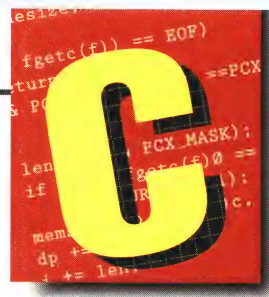
Raima Deutschland: 49 7022 9256 10 Italy: 39 49 80 77 140, 39 49 720 577 Argentina: 54 1 470 72 03
Raima UK: 44 1 273 819 292 Raima Benelux: 31 35 694 4738 France: 33 1 446 100 42
Baltic States: 372 639 8702 Finland: 358 9 8045 130 Russia: 7 812 316 1965 Japan: 81 3 3980 2235 Malaysia: 60 3 262 98 68

Document: Done

www.raima.com

C for C++ Programmers

Al Stevens



After reading this column, you might be tempted to accuse me of exercising the Pournellian imperative by using my column to promote my own books. Yes, I am going to discuss some books that I wrote. No, the typical reader of this column, who already knows C and C++, does not need to buy or read any of these books. Therefore, these plugs—er, discussions—won't serve to increase my royalties by any significant amount. They do, however, lead to some discoveries about C and C++ that I will share with you.

It started early this spring when I began work on a fifth edition of an introductory C++ book called *Teach Yourself C++*. I've never liked that title, which the publisher chose for the first edition in 1990 after I rejected *C++ for C Programmers*, which Ira Pohl had already used. I always considered the title "Teach Yourself" to be only a notch higher on the schlock scale than something like *Quantum Electrodynamics for Dummies*. Someone must have liked it, however, because at least two other publishers have published books with the same dumb title. Title notwithstanding, the work has been professionally and financially rewarding, and I am proud of the content, particularly as it has changed and grown since the first edition. The fifth edition is, I think, the best.

If (Editions == 5) Why?

There are several reasons to upgrade a book that addresses technology—the technology changes, the author's knowl-

edge changes, the potential readership changes, or some combination thereof. All these reasons apply to this new edition of *TYC++*, and I'll discuss them and address some things about C++ that I discovered as I modified major parts of the book.

Writing a new edition of a programming language book means starting with the previous edition, going through it chapter by chapter, exercise by exercise, and reviewing what has changed in the language. There's probably no reason to revise a language book if the language hasn't changed, other than the usual publisher avarice and rapacity. (Don't worry. I'm not in any trouble. Publishers tend not to know words like that, and editors don't bother telling them.) The C++ draft standard has undergone only minor changes since the publication of the fourth edition of *TYC++*, but the C++ language being used has seen dramatic differences, primarily because more and more compilers now implement recent language inventions (also called innovations) of the ANSI committee. From my perspective, the so-called standard language is, at any given time, what is available and acceptable to programmers, rather than what is under deliberation. The final (we hope) draft standard document is still being considered, and no compiler has yet implemented the complete standard language as currently defined. Consequently, even though the fifth edition of *TYC++* is still in the works, I have already agreed to write a sixth. Watch for it around Christmas of 1999, irrespective of the committee's status and the compilers' compliance. Whatever else happens, the C++ of late 1999 shall be the C++ of record as far as this author is

concerned. At least that's what I'm saying now.

Changes in my understanding of the language and how it ought to be used influenced the fifth edition, too. Over the years, the book's content and my knowledge of C++ have been affected by the comments of readers and C++ experts. Greg Comeau of Comeau Computing, who knows a lot more about C++ than most of us, generously spent some time and marked up a copy of an early edition to correct some errors. John Dlugosz offered suggestions in areas where my code worked with contemporary compilers but did not conform to acknowledged C++ conventions or proposed standards. Those are only two of the many programmers who helped me. Since 1990, I have read many books on C++ and learned from most of them. My own programming experience on several different platforms—DOS, Windows 95, Windows NT, OS/2, UNIX—has, over the years, changed my views of how C++ code ought to be written. All these things and all these people change what I understand and, consequently, what I write.

Finally, changes in the kinds of programmers who need a book like this—its targeted audience— affect its content. All the previous editions, like most introductory books on C++, assume that the reader is a C programmer. I wrote a companion book, called *Al Stevens Teaches C* (another publisher-suggested title that I came to hate), for programmers who do not yet know C.

Since then, I have discovered two things. First, the C component of C++ has significant differences from ANSI C. (Not

Al is a DDJ contributing editor. He can be contacted at astevens@ddj.com.

necessarily news when regarded in fragments, but substantial when viewed as a whole.) Some of these differences are found in the language itself; others are reflected in how C++ programmers use the language. Second, most C programmers have already made the move to C++. There is not nearly as much need for a C++ book that assumes a C programmer as there used to be. Potential readers of an introductory C++ book today are programmers who know neither language.

Consequently, I decided to adapt some of the C book's contents into the first several chapters of the C++ book, turning *TYC++* into a book that covers the subject wall to wall. And that's where I made some interesting discoveries about

C from a C++ perspective. I think that most programmers who ascended from C into C++ know and understand these things, but viewing them as a group provides a dramatic insight into how C++ has influenced its C component. And viewing them from the surrogate perspective of someone who knows neither language and is learning C++, which the author of such a tutorial must do, is a revealing exercise. I offer these observations, conclusions, and opinions here for anyone who sets out to teach C++ with or without my book.

C++[C]==--C;

First off, there is a lot about C that you might not need to teach. For starters, the

whole *stdio* family of functions, although included in C++, has become unnecessary, having been replaced by the C++ stream classes. What's the point of teaching two ways to do something when the contemporary way is clearly superior? Remember, you are teaching C++—not C—to a programmer. It's sufficient to refer those who are interested in ANSI C to an ANSI C textbook. Why dwell on *malloc*, *calloc*, and *free* when the C++ *new* and *delete* operators do a better job? You can bypass teaching *setjmp* and *longjmp*, too, because C++ exception handling does the same thing only better. You don't have to mention NULL because C++ programmers tend not to use it. Certainly, you can leave out the old-style C function parameter declarators. C++ compilers don't support them. You don't have to explain the need for *void* in a prototype's empty parameter list, because C++, although it permits a *void* parameter list, does not need the *void* keyword to distinguish a prototype from an old-style C function declaration; C++ does not support the classic K&R C function declaration. C typecasts are old hat now that we have C++ *new*-style casting. The *string.h* functions are candidates for the junk pile, being replaced by the more intuitive C++ *string* class.

After much thought, I decided to teach some of the obsolete (from a C++ programmer's perspective) C idioms and briefly touch on the Standard C library's *string*, memory allocation, and *setjmp* functions if only because a C++ programmer is likely to run into them. Sometimes the C string functions (*strcpy*, *strcmp*, and the like) are more efficient than the C++ *string* class, which can make a difference in a time-critical application. I found *setjmp* and *longjmp* particularly useful as a precursor to C++ exception handling because they do explicitly what the C++ *try*, *catch*, and *throw* operators do under the surface. So, this particular obsolete C behavior turns out to have a lingering use after all. I also decided to address C casts during an exercise early in the book. The exercise needed a cast and the book was nowhere near an advanced-enough stage to teach the appropriate *new*-style cast. Finally, if you are going to teach a programmer to overload global *new* and *delete*, it's easier if you start by teaching *malloc* and *free*.

CPP = Forever

You cannot, however, ignore the preprocessor, even though many C++ luminaries would like to see the preprocessor quietly become extinct. Inline functions are an improved way to write macros with parameters, and *const* variables replace



Game DEVELOPER
ON THE FRONT LINE OF GAME INNOVATION

We're New!

Are You on the Front Line of Game Innovation?

You are if you're reading *Game Developer* magazine.

We got a makeover, we've gone monthly, we're free if you qualify—and we're still way deep!

Find even more hardcore technical coverage on programming, 3D and visual design, sound design, and game design strategy in *Game Developer*. It has everything your entire professional team needs to bring killer games to market.

Game Developer is FREE to any member of the professional game development team!

Call 800.444.4881 or stop by www.gdmag.com and fill out the qualification form for your free subscription today!

#defined global symbols, but these improvements do not totally replace macros, particularly when certain macro constructs provide mnemonic expressions that expand into complex initialized data structures. Prominent examples are the macros that GUI application-framework class libraries use to manage things such as message maps. And consider the trace macro

```
#define trace(p) cerr << #p " = " << p << endl
```

that I use to avoid launching the cantankerous gdb debugger. You can't do that with inline functions. You must teach the preprocessor because, sooner or later, most C++ programmers need it.

If (C++ > C) ...

Most introductory C++ books include chapters somewhere near the beginning that discuss the differences between C and C++ with respect to notational improvements that C++ makes. Examples of these are the scope resolution operator, default function arguments, unnamed function parameters, the fact that function prototypes are required, variable declaration placement, inline functions, anonymous unions, variable declaration inside of *for*-loop and *if*-condition expressions, double-slash (//) comments, *const* instead of *#define*, *structs* as types, references, function overloading, and the *signed* type specifier.

An introductory C++ course that does not assume a student's knowledge of C presents a quandary. These subjects are best addressed in the C++ context in which they occur rather than as C improvements. The explanations of, for example, unnamed function parameters and default function arguments should accompany the discussions on function parameters and arguments—as an integral part of the C++ language rather than as an extension to C. So what's the quandary? Just this. The student who, coincidentally, happens to already know C, may be tempted to sleep through—or, in the case of a book, skip over—the function parameter/argument discussion

```
void foo(struct Date yesterday)
{
    // ...
    goto bypass;
    {
        struct Date today = yesterday;
        // ...
    bypass:
        // ...
    }
}
```

Example 1: Valid C, invalid C++, *goto* usage.

and, by doing so, misses learning about the improvements. What to do? You have no choice but to tell them they have to stay awake and alert through it all. But a student can lose interest wading through so much familiar material in wait of that occasional gem of new information.

Here are some more subtle differences that a C programmer might miss by skipping over the basics.

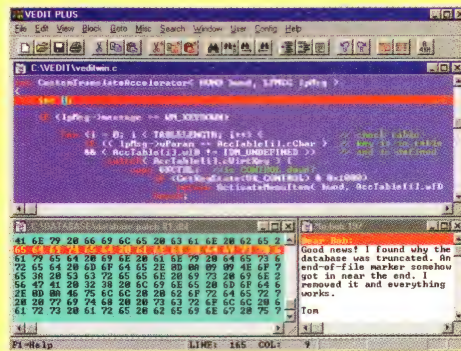
- A void pointer in C++ cannot be assigned to a pointer to a type without a cast.
- The C++ *main* function cannot be called recursively in a C++ program. C has no such restriction, although it probably should.

- You must initialize a *const* object in C++.
- Variables cannot be implicitly declared as *int*.
- An enumerator is a type, not an *int*.
- *wchar_t* and *bool* are real types, not *typedefs*.
- *floats* are not always promoted to *doubles* in expressions.

As an author, I'm not sure I can solve the problem of a reader's attention span in such a book—short of putting those eye-catching, tacky Dummies-like icons in the margins: "Stop!" "Look here!" "Important!" "New stuff!" "Wake up!" Such a practice would ruin my book's chances of becoming a respected textbook in the hallowed halls of higher learning. Wait a

New VEDIT PLUS 5.0 for Windows.

The solution to all your file editing, patching, translation and conversion problems.



- Program development
- Text processing
- CD-ROM development
- Database maintenance
- Mainframe - PC conversions
- "Year 2000" projects

Originally created in 1980 as the first programmer's editor, the new VEDIT PLUS 5.0 has evolved into a unique multi-function tool that effortlessly handles all of your routine, tricky and huge file editing needs.

Advanced features include multi-file, multi-window editing, template editing, syntax highlighting, search/replace with regular expressions and columnar blocks. Fully configurable keyboard and emulations.

Only VEDIT is fast enough for huge 100+ Megabyte files. It can perform thousands of search/replace operations in little more time than it takes to just copy the file. (Other editors can take days for a single global search/replace.)

Written mostly in assembly language, it's the most responsive program you'll ever use. Instant startup, instant everything.

VEDIT PLUS - Windows/DOS single user license: \$149. (Includes both Windows and DOS versions for Windows 3.1, 95 and NT)

Visit our extensive Web site at
<http://www.vedit.com>.

Lightning fast editor for database, CD-ROM, mainframe & all huge files

- Edit **absolutely any file** - text, data, or binary up to 2 Gigabytes.
- Edit in ASCII, Hexadecimal, Octal, EBCDIC or any combination of modes.
- Edit DOS, Unix and Mac text files, data files with fixed / variable length records.
- Translate between EBCDIC and ASCII, IBM PC and ANSI. Flexible sorting.
- Powerful C-like macro language for writing custom translators, filters, more.

New VEDIT PLUS 5.0 features

- Windows version is exceedingly fast & compact. (220K .EXE file, no .DLLs)
- Windows 95 long filename support. (Both Windows and DOS versions.)
- Toolbar, right mouse button support, keyboard shortcuts, configurable.
- Improved word processing, printing, indenting and much more.

VEDIT is a registered trademark of Greenview Data Inc.

Toll Free: 1-800-45-VEDIT (1-800-458-3348)
Telephone: (313) 996-1300, Fax: (313) 996-1308
E-Mail: sales@vedit.com, Web: <http://www.vedit.com>
Mail: P.O. Box 1586, Ann Arbor, MI 48106

Greenview Data

minute! What am I saying? What faculty in its right mind would adopt into its curriculum a textbook titled *Teach Yourself* anything? The students and the administration would ask the teachers, "What do we need you for?" Now I know why I don't like that title.

Believe it or not, the much maligned *goto* flow-control statement works quite differently in C++ than it does in C. You might wonder why they bother fixing *goto* when everyone tells everyone else not to use it. But *goto*, as permitted in C, would have a dangerous side effect in C++. Consider Example 1, which is valid C code and invalid C++ code. C++ does not permit *goto* to jump around the initialization of an automatic object, which could bypass a necessary constructor.

Class == Structure ? Not : Almost

C++ defines structures and classes almost identically. The only differences are related to access specifiers and inherited access. The members of a structure have public access by default. The members of a class have private access by default. Classes derived from a structure are derived publicly by default. Classes derived from a class are derived privately by default. If you didn't already know that, then go buy my book. Please?

This is not a difficult concept to explain when you are teaching C++ to a C programmer. You can ignore *structs* for a while and teach the behavior of classes. When the student understands classes, you can reveal the surprise that *structs* and classes are almost identical with the only differences being those I just explained. The C programmer has no problem with that. In fact, many C programmers are never taught that structures and classes are alike, and they program for years believing that C++ *structs* are just like C *structs* with no member functions, access specifiers, or participation in inheritance.

But when you are teaching C++ from scratch, the question that begs to be answered is this: Why have two different constructs when the differences are so small? I suppose you could ignore *structs*, but that would bypass necessary information; the programmer is bound to encounter them in C++ programs written by others. The only way to understand the reasons for the two is to understand that C++, as it strives for compatibility, inherits a lot of language from C. Which means that you must teach not only the origins and history of C++, but the consequences of all that legacy baggage, too.

The answer to the *class* versus *struct* question goes back to when the C++ class

was evolving from the C *struct*. There is no real technical reason why the C++ *struct* should include all the properties of the class. C++ cannot eliminate the *struct* altogether; it must support the C *struct* to preserve compatibility between the languages according to Bjarne Stroustrup's original objectives for the C++ language. But without an explanation from Stroustrup, it is not clear why the *struct* needs member functions, access

Many C programmers are never taught that structures and classes are alike

specifiers, and the ability to participate in class hierarchies. In *The Design and Evolution of C++*, (Addison-Wesley, 1994), Stroustrup says:

My intent was to have a single concept: a single set of layout rules, a single set of lookup rules, a single set of resolution rules, etc....Only a single concept would support my ideas of a smooth and gradual transition from "traditional C-style programming," through data abstraction, to object-oriented programming.

So, the reason is cultural rather than technical. Stroustrup goes on to explain that keeping the *class* and *struct* the same forestalled an otherwise unavoidable tendency on the part of language standardizers and specifiers to overwhelm the *class* specification with excess features while leaving the *struct* to implement only those features that involve low overhead and simplicity.

Given all that, when should you use a *struct* rather than a *class*? Or should you ever? Many programmers adopt this rule:

When the data structure's implementation is the same as its interface, use a *struct*. Otherwise use a *class*.

What are the implications of this rule? What does it mean when you say that the implementation is the interface? I developed the strategy several years ago

for my own programs, but I never heard it expressed so well. I got the phrase from *Ruminations on C++* (Addison-Wesley, 1997), by Andrew Koenig and Barbara Moo. This is a nice book with cover art that hides some surprises. At first it looks like the cover for a paperback edition of *The Yearling* or *Huckleberry Finn*. Then, because of the black and white cows, the book looks like a Gateway 2000 manual. Why the cows, I wondered. Then it hit me—one of the authors is named "Moo." Then I remembered that "ruminating" means "chewing cud" as well as thinking about stuff. The book is an edited collection of Koenig's columns in various publications related to C++. Although some of his ruminations are dated (Old cud?), reflecting their age and the technology as it existed when he wrote them, this book is well worth a careful read. I highly recommend it.

Saying that a *struct*'s implementation is its interface means simply the *struct* is typical C with data members only (no member functions), all of which have public access so that the format of the data members is what the application program views.

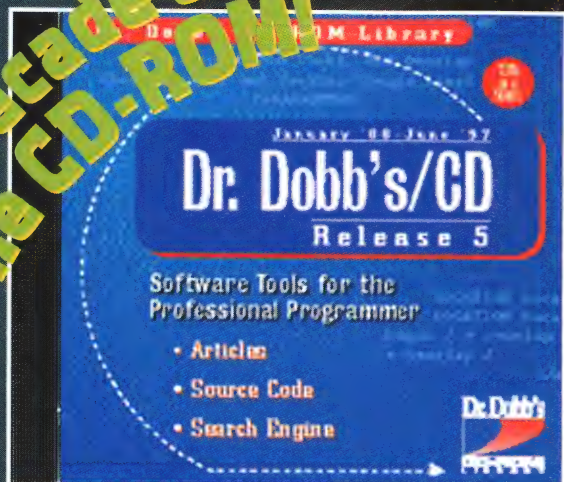
Object-oriented purists would contend that by following that rule, there is no valid application for the C++ *struct*. If you tend to fall into that camp, then C++ supports your beliefs. Use classes exclusively. If you lean more toward the pragmatic approach, then C++ permits you to write code by using either idiom. C++ has something for everyone.

C = C > C++ ? C : C++

C++ is sometimes called a superset of C, which makes C a subset of C++. Except that Standard C came first. C++ has been called an object-oriented extension of C, but that doesn't seem to work either because, as I learned, the C that C++ supports is not the C that Standard C compilers support. If you can forget about the chronology, perhaps the best description of the relationship between the two languages can be expressed like this: If you are comparing Standard C to C++, they are two different languages. The C component of C++ and the programming models that it supports are different enough from Standard C to set them apart. If you are addressing the C component that C++ compilers support, perhaps the best way to describe it is as a dialect of C++, one that enjoys certain notational improvements over Standard C and leaves out the data abstraction and object-oriented idioms. Stroustrup's "better C." C as a dialect of C++. What a peculiar notion!

DDJ

Over A Decade of Dr. Dobb's
On One CD-ROM!



**NEW
RELEASE!**

Visit the Dobb's Web site! www.ddj.com

Dr. Dobb's CD Release 5 provides you with immediate and complete access to nine and a half years of *Dr. Dobb's Journal*. Every Algorithm Alley, every Swaine's Flame, every article, and every bit of source code from January 1988 through June 1997 is included on this CD-ROM. In addition, *Dr. Dobb's/CD Release 5* contains all issues of *Dr. Dobb's Sourcebook*. The magazine

devoted to spotting and analyzing emerging trends in software technology.

Bringing together a decade's worth of perspectives and expertise from the most respected programmers in the industry, *Dr. Dobb's CD/Release 5* offers both longtime and first-time readers an unparalleled programming resource. Use its powerful search engine to find what you are looking for and then print it, copy/paste it, or directly export it as a file.

No more sifting through dozens of back issues and manually keying source code into your current programming project. With *Dr. Dobb's/CD Release 5*, you have it all when, where, and how you want it!

Super-fast Full Text, Boolean, Search Engine

Indexes for both Articles and Authors

Complete Printing Access

Copy/Paste Functionality

Export Files

Annotate and Bookmark for Your Future Reference Needs

Save Searches for Quick Access Later

\$29.95 upgrade available to prior owners! To upgrade call:
800-822-1162

Call Now! 800-922-0549

E-mail: orders@mfi.com Fax: 913-841-2624

Mail: *Dr. Dobb's CD-ROM Library*
1601 West 23rd, Ste. 200
Lawrence, KS 66046-2703

International: Use mail, fax, e-mail, or call 913-841-1631

YES! Send me the Dr. Dobb's/CD Release 5 CD-ROM today!

Dr. Dobb's/CD Release 5 CD-ROM

\$ 49.95

Please add sales tax in the following states: CA (8.5%), GA (6%), IL (6.25%), KS (6.9%), NY (8.25%), TX (8.25%), Canada (7% GST).

\$ _____

Shipping (\$2.00 for each CD-ROM shipped in USA/Canada, \$10.00 for Airmail to all other countries).

\$ _____

Total

\$ _____

Name _____

Address _____

City/State/Zip/Country _____

Phone _____

E-mail _____

Here's how I'm paying (check one): ☐ VISA ☐ MC ☐ AMEX ☐ Check enclosed

Credit Card number (include all digits) _____

Exp. date _____

Signature required _____

Please allow 2 weeks for delivery in the USA/Canada; 4 weeks for all other countries, checks/money orders must be drawn on a U.S. bank in U.S. funds only

J0797

Dr. Dobb's
CD-ROM
LIBRARY

RELEASE 5 INCLUDES THESE MUST-HAVE FEATURES:



**Our competition may have teeth,
but we rule in Java™ tools.**



Visual Café & Visual Café Pro

Symantec® Java tools. They've won every major industry award. A #1 ranking from the experts. And the acclaim of Java developers worldwide. Combining drag-and-drop ease with robust coding tools, Visual Café™ is "the new champion of Java development environments" (Windows Magazine), the "gold standard by which other RAD tools should be judged" (Information Week). And for database apps, new Visual Café Pro™ is already "one step ahead of the others" (InfoWorld). Sure, there are other fish in the sea. But for Java tools that rule, there's only one choice: Symantec.



Symantec vs. the "Big Fish": you decide.

	Symantec®	Microsoft®
Visual drag and drop form design / RAD	yes	no
Library of pre-built Java components	yes	no
Live Java Layout Manager support	yes	no
Interaction Wizard	yes	no
JDBC Support	yes	no
Support for Oracle, Informix, Sybase, ODBC	yes	no
Windows 95®/Windows NT® support	yes	yes
Macintosh® support	yes	no
JDK 1.1 and Java Beans support	yes	no

Call for a FREE trial copy!
1-800-453-1059 ext 9A34
Or download from: www.cafe.symantec.com



SYMANTEC.

Symantec and the Symantec logo are registered trademarks and Symantec Visual Café and Symantec Visual Café Pro are trademarks of Symantec Corporation. Java is a trademark of Sun Microsystems, Inc. Microsoft and Windows are registered trademarks of Microsoft Corporation. All other brand names or trademarks are the property of their respective owners. ©1997 Symantec Corporation. All rights reserved. In Canada, call 1-800-365-8641. In Australia, call 02-879-6577. In Europe, call 31-71-535-3111. Free trial copies of Visual Café only, while supplies last.

How do I Write an International Application?

Cliff Berg



The Internet has opened up the world to computing. Simply by using e-mail, or transferring/hosting software on the Web, it becomes as easy to work with people on the other side of the globe, as it is to work with people on the other side of town. Java helps make this possible because it is a network-capable language. International companies want to take advantage of this boundless computing model, and write applications that customers and a global workforce can access anywhere—in any language.

Java is touted as a multilanguage-capable computer language because it provides built-in support for Unicode. This is not enough, however, for true multilanguage support. There also needs to be support for displaying monetary currencies, and foreign number, date, and time formats. It would also be nice if multilanguage support was easy to implement, without hardcoding of language-specific strings. JDK 1.1 provides a suite of packages for multilanguage and multinational development. While I won't get into all of the features of those packages here, I will show how to use some of the more important ones.

Locales, Resources, and Resource Bundles

JDK 1.1 introduces the term “locale”—a specification of any of spoken language, geographic location, and additional qualifiers that may be built-in. Generally, a locale can be, for example, the combination “English” and “England,” or “English”

and “Australia.” A locale need not specify everything: The resource that most closely matches the specification is used, and if none matches, resources for a default locale are used.

The new JDK 1.1 resource mechanism (in the package called “util”) is the foundation for the internationalization facilities. Resources are location-independent collections of information that may be retrieved by name. They relocate automatically with the program no matter how it is deployed. A resource can be an image file, text string, or any Java object. It may be stored in a property file or as a stream of bytes.

The key feature of a resource is that it is designed to follow the application. It may be encapsulated into a JAR file, for example. Regardless, it is always accessed the same way—with the class loader's *getResource()* method. (A *Class* object also has a *getResource()* method, but it merely calls the *getResource()* method of the class loader that loaded the class.)

A *ResourceBundle* is a collection of resources specifically designed to aggregate those resources needed for a specific geographic locale or spoken language. A naming convention serves to identify specific-resource bundles according to their locale, so that the resource-bundle methods know which resource bundle to select, based on the current locale. A locale can be set on a per-object basis, as opposed to a system-wide basis, making it possible to deal with more than one locale at a time. This is important, for example, for multilanguage programs, such as language translators.

The Calculator Class

Figures 1 and 2 show multilanguage desk calculators that let you select from one of two languages (German and English),

and one of two modes (Standard and Currency). Whenever one of these selections is changed, the calculator reconfigures itself accordingly. For example, when German is selected as the language (as in Figure 1), control labels are displayed in the German language. If mode is set to Currency, numeric values are displayed as currency values, either in dollars or Deutschmarks, depending on the language selected. Figure 2 shows an English version of the calculator.

The *calculator* class (available electronically; see “Availability,” page 3) is implemented as an applet, since one use of an international program is as a browser-based application. Since JDK 1.1 is required, browsers that have not yet implemented JDK 1.1 will not be able to run it. At this writing, HotJava and Appletviewer support JDK 1.1.

The first thing the applet does is instantiate all of its GUI components in its *init()* method. Since this is a 1.1 application, each component specifies an event handler (an *ActionListener* or *ItemListener*, in this case). For simplicity, I have chosen to put all of the event-handling behavior in the applet, so the applet implements both *ActionListener* and *ItemListener*. The *init()* method then calls *updateGui()*, which obtains the resources needed to support the new settings, then redisplay the applet with those settings and resources. This method is also called whenever users reconfigure the applet by changing the language selection, or the calculator mode.

The first thing *updateGui()* does is call *getLocaleSetting()*. This method gets the user's language selection by comparing the text of the language-choice selection with the built-in language name string for each of our two supported languages.

Cliff, vice president of technology of Digital Focus, can be contacted at cliffbdf@digitalfocus.com. To submit questions, check out the Java Developer FAQ web site at <http://www.digitalfocus.com/faq/>.

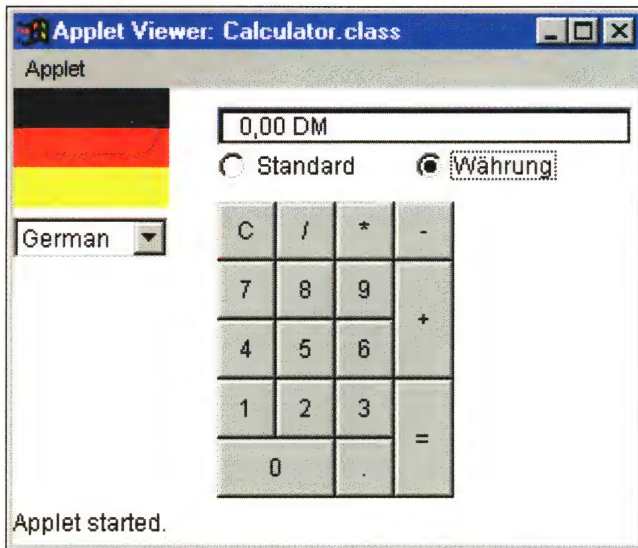


Figure 1: Calculator applet set up for German currency.

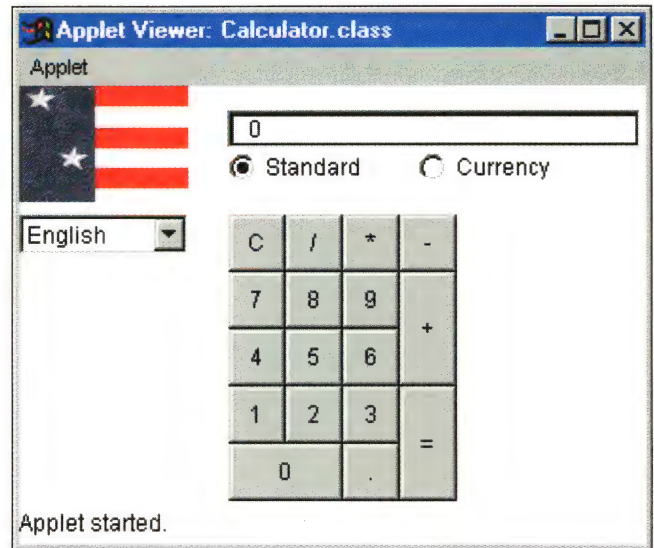


Figure 2: Calculator applet set up in English.

These are the same strings used when building the language-choice control, so one should match the user's selection. When a match is found, the private variable "locale" is set to the corresponding built-in locale. (There are presently 28 built-in locales.) `getLocaleSetting()` then gets the resource bundle for the locale by calling the static method `java.util.ResourceBundle.getBundle("ButtonLabelResources", locale)`. This method attempts to locate the `ButtonLabelResources` resource bundle for the specified locale. A resource bundle can be a property file or Java class. I have chosen to use a property file in this case. (In fact, there are two property files—`ButtonLabelResources.properties`, and `ButtonLabelResources_de.properties`.) The `getBundle()` method uses the convention "<bundle_name>_<language>_<country>.properties" to identify a resource file, and since "de" identifies German, the German language resource bundle will automatically be loaded. I could have specified a country (there are currently three German-speaking countries defined), but chose not to, so only the language will be used as a resource selection criteria. (At this writing, the supported locales are listed at <http://www.javasoft.com/products/jdk/1.1/intl/html/intlspec.doc2.html>.) Finally, `getLocaleSetting()` sets the current locale to be the one that the user has selected.

The `updateGui()` method makes the calls in Listing One (listings begin on page 116). To understand these lines, look at what is contained in the aforementioned properties files. The standard properties file (the one without `_de` in its name) contains Listing Two. Thus, three properties are specified. If the system chooses this properties file when it searches for the

`ButtonLabelBundle` resource bundle, it will get these values. If on the other hand it chooses the German resource bundle, contained in `ButtonLabelBundle_de.properties`, it will get the values in Listing Three.

The first line contains a Unicode-escaped character sequence, which represents the German word for currency (Währung). If I had a German-language Unicode-based editor, I could have typed this word in directly, without resorting to the escape sequence.

The first line in each property file is, therefore, the label that is to be used on the radio button that the user clicks when selecting currency mode. When German is selected as the locale, this is displayed in German; when working in English mode, the word "Currency" is displayed.

The second line is the label to use for the other radio button, which selects non-currency, or Standard mode. Since the German adjective for "standard" is the same as the English word "standard," the second line in each file is identical. The last line in each file specifies a GIF filename, which is loaded by Listing Four.

I get the name of the file resource from the resource bundle, then use that name to find the actual resource—the file. I then request a repaint. Since my `paint()` method draws the `imageURL` object on the applet, I effectively have a different image drawn for each language selection. Note that I display the selected country's flag.

Finally, `updateGui()` makes the calls in Listing Five in which the first line sets a format-object reference depending on the state of the mode checkboxes. A format object (in the package called "java.text") can be one of several kinds of object that act as input and output formatting services (conceptually similar to the way C-

formatted I/O functions work—although better thought out). I use a numeric formatter when the mode is Standard, and a currency formatter when the mode is Currency. Note that specifying the locale is sufficient for the formatter to know how to format numbers and currency amounts uniquely for each locale. (In many European countries, thousands are separated with periods, and the decimal point is displayed with a comma—exactly the reverse of the standard practice in the U.S.) The formatting is even smart enough to know that the German currency is Deutschmarks, but alas, not smart enough to call an online currency-exchange service to get the going conversion rate. Listing Six is the code that runs the applet.

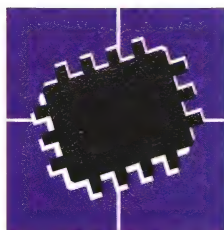
The complete source code for the applet is available electronically (see "Availability," page 3) and at the Digital Focus web site (<http://www.digitalfocus.com/ddj/code/>).

Conclusion

The internationalization resource mechanisms provided by JDK 1.1 are not only flexible and powerful, but extremely easy to use. There are a large number of methods, only a few of which I've touched on here. Do not be intimidated, however, because the API is well thought out and simple to understand. Furthermore, once you get used to the resource mechanism, you will find that it makes keeping track of images and other items used by a program much easier. You'll be able to package resources into a JAR file with your applet and not have to worry about whether the applet can find them.

DDJ
(Listings begin on page 116.)

**Everyday you're asked for the impossible.
Everyday you give it.**
Sometimes even superheroes need a little



**EMBEDDED
SYSTEMS
CONFERENCE**

**Embedded Systems Conference
Sept. 29 - Oct. 2, 1997
San Jose, CA**

**Come to the technical conference for
embedded design superheroes.**

Four days of new ideas, new products,
practical skills, and engineer camaraderie.

More than 120 classes!

- Debugging ■ C/C++
- Verification/Validation
- Web Connectivity ■ Java

Thousands of products on display

- Microprocessors/Controllers
- Development Tools
- Emulators
- Operating Systems

Call for your free course catalog!

415-278-5259

Fax: 415-278-5200

Email: esc@mfi.com

World wide web: www.embedded.com

Send me the details!

I am interested in: ☐ Attending ☐ Exhibiting
☐ Receiving conference e-mail updates

Name

Title

Company

Address

City

State

Zip

Phone

Fax

E-Mail

Embedded Systems Conference • Miller Freeman, Inc.
525 Market Street, Suite 500 San Francisco, CA 94105

DJ797

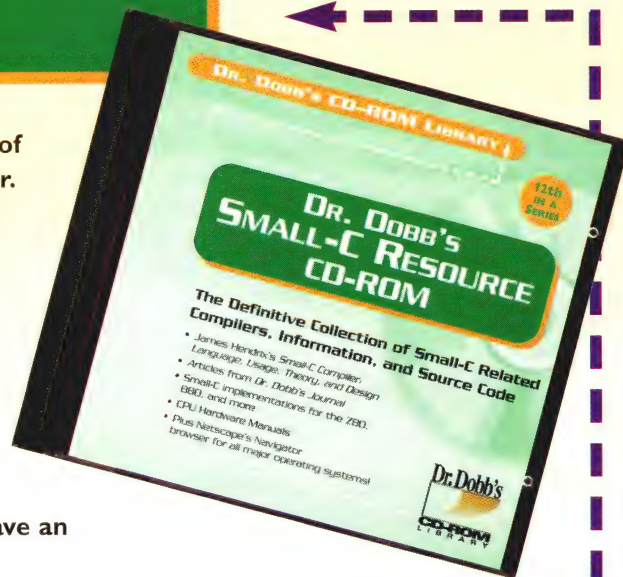
DR. DOBB'S SMALL-C RESOURCE CD-ROM

NEW!

The *Small-C Resource CD-ROM* contains the definitive collection of compilers, information, and source code for the Small-C compiler. Packed with books, tools, utilities, and Small-C implementations, this CD-ROM belongs in every serious programmer's collection.

This CD-ROM includes James Hendrix's out-of-print book, *A Small-C Compiler: Language, Usage, Theory, and Design* and several relevant articles from *Dr. Dobb's Journal* written by C experts such as Allen Holub.

Plus all content is hyperlinked in HTML format, which makes viewing easy and convenient. And for those who don't have an HTML browser, included is Netscape's Navigator browser supporting all major operating systems.



Visit our website
for more information at —
www.ddj.com/cdrom/

To order:

Call: 800-992-0549

Fax: 913-841-2624

E-mail: orders@mfi.com

Int'l: Use mail, fax, e-mail, or call
913-841-1631

Mail orders:

Dr. Dobb's CD-ROM Library
1601 West 23rd Street, Ste. 200
Lawrence, KS 66046-2700 USA



Send me the *Dr. Dobb's Small-C Resource CD-ROM* immediately. I will also receive the *Dr. Dobb's Sourcebook, Distributed Objects* as a bonus. I understand that I have a full money-back guarantee.

Small-C Resource CD-ROM.....\$49.95

Please add sales tax in the following states: CA (8.5%), GA (6%), IL (6.25%), KS (6.9%), NY (8.25%), TX (8.25%), Canada (7% GST).

\$ _____

Shipping (\$2.00 for each CD-ROM shipped in USA/Canada, \$10.00 for Airmail to all other countries.)

\$ _____

TOTAL AMOUNT.....

\$ _____

Name _____

Address _____

City/State/Zip _____

Country _____

Phone _____

E-mail _____

Here's how I'm paying (circle one): VISA - MC - AMEX - Check enclosed

Credit card number (include all digits) _____

Expiration Date _____

Signature required _____

Please allow 2 weeks for delivery in the U.S./Canada; 4 weeks for all other countries. Checks/money orders must be drawn on a U.S. bank in U.S. funds only.

DJ97

Place your order today!

ONLY \$49.95

Dr. Dobb's
CD-ROM
LIBRARY

Decimating Polygon Meshes

Will Schroeder and Tom Citriniti

Computer-generated, interactive 3-D graphics is growing in importance and application. Many games and even sophisticated user interfaces are based on 3-D graphics technology. For example, *Doom* and *Quake* use 3-D graphics in combination with texture maps to create vivid, adrenaline-producing games. Web-based technology, such as VRML, enables users to fly through virtual worlds or interact with databases. The use of 3-D graphics is likely to explode as new software tools and graphics accelerators are introduced into the marketplace.

Unfortunately, a limiting factor in the use of 3-D graphics is the amount of data that must be processed, transmitted, and rendered. In our work environment, we bump into this limitation regularly: One of our applications involves visualizing the internals of aircraft engines and other mechanical devices. In this environment the database ranges in size from 10–100 million triangles, with the desired goal of flying through the data at 3–30 frames per second. You may have experienced similar limitations if you've ever tried to use VRML applications or to view VRML content on the Web. Many of these applications are far from interactive.

To address this problem, a relatively new field has emerged in computer graphics, often referred to as "triangle decimation" or "polygon reduction." The aim of these techniques is to reduce the number of polygons in a mesh while maintaining a good approximation of the original data. The focus of these algorithms is on polygons—especially trian-

gles—because 3-D graphics accelerators are designed to process these geometric primitives. Triangles are used because surface primitives, such as polygons or splines, are triangulated before being handed off to the graphics hardware.

In this article, we will focus on a decimation algorithm that is fast, widely used, and for which source code is available. We will also show how it is used, and how to create a decimated VRML file.

Polygon Reduction Algorithms

Polygon reduction algorithms can be classified in two ways: whether they modify the geometry of a mesh, and whether they modify the topology of the mesh. When we say "mesh geometry," we mean the x-, y-, and z-coordinates of the vertices, while mesh topology refers to features like holes or handles (imagine a torus). Algorithms that modify the geometry of a mesh will create new vertices, or possibly move existing vertices to new positions. The advantage of modifying geometry is that vertices can be repositioned to give a better approximation of the original mesh. The disadvantage is that if attribute information like texture coordinates or color is associated with the vertices, it must be mapped to new values when the vertex location changes. Unfortunately, both the calculation of new vertex positions and attribute mapping tend to be expensive operations.

Algorithms that modify topology will close holes (imagine removing rivet holes in sheet metal), add holes or split the mesh, or even introduce nonmanifold attachments. An example of a nonmanifold edge is an edge shared by three or more triangles, rather than the usual two (interior edge) or one (boundary edge).

Most polygon reduction algorithms will preserve the topology of a mesh, but it

is common to modify the mesh geometry. The decimation algorithm described here preserves both the geometry and topology of a mesh.

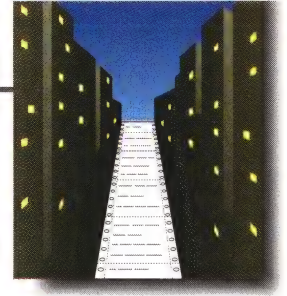
The Decimation Algorithm

The decimation algorithm performs a series of local operations on a mesh to gradually reduce the number of triangles and vertices. These operations are performed until the desired reduction is achieved, or until no more operations are possible due to topological or other constraints. As Figure 1 illustrates, there are three major steps in the process:

1. Classify a vertex according to its local geometry and topology. Classification also involves estimating the potential error introduced into the mesh if the vertex is deleted in steps 2 and 3.
2. Delete the vertex if it is of appropriate classification and if its error measure is less than some threshold value. Vertex deletion causes the removal of all triangles using the deleted vertex, and results in a "hole" in the mesh.
3. Patch the hole with a new triangulation. The net effect is two fewer triangles if the vertex is in the interior of the mesh; or one less triangle if the vertex is on the boundary.

The classification of a vertex depends upon both the local geometry and topology. For example, vertices that have an edge that is used by more than two triangles are considered nonmanifold. This is a topological classification. Feature edges are used for geometric classification. Feature edges are found when the normals of the two triangles on each side of the edge form an angle greater than a user-specified feature angle.

Vertex classification is important because, if done correctly, it can help preserve sharp edges and corners (edges



Will, a computational scientist at GE's R&D Center, can be contacted at schroeder@crd.ge.com. Tom, an adjunct assistant professor at Rensselaer Polytechnic Institute, can be reached at citrit@rpi.edu.

and corners of a cube, for example). Feature edges are identified and the triangulation of the hole is controlled to preserve these sharp edges. Also, certain vertices, like nonmanifold vertices, are not deleted to ensure that the topology of the mesh is not modified.

An important feature of this algorithm is how the error is estimated. One simple measure of error is computing the distance from the original mesh to the decimated mesh after vertex deletion (es-

entially a measure of distance to plane). Obviously, if the triangles around a vertex all lie in the same plane, then the deletion of the vertex introduces no error. Or, if the vertex lies on the boundary of the mesh, the error measure is the distance of the new mesh edge to the deleted vertex (distance to edge). For more information on this algorithm, see "Decimation Of Triangle Meshes," by W.J. Schroeder, J. Zarge, W. E. Lorensen, *Proceedings of Siggraph '91*.

Applications

To demonstrate the application of decimation, we'll use the Visualization Toolkit (vtk) available at <http://www.cs.rpi.edu/~martink/>. Vtk is a C++ class library with a built-in Tcl/Tk interpreter for visualization and 3-D graphics, and includes source code, data, and many examples. The software runs on UNIX and Windows 95/NT. For more information, refer to *The Visualization Toolkit: An Object-Oriented Approach to 3-D Graphics*, by Will Schroeder, Ken Martin, and Bill Lorensen (Prentice-Hall, 1996). [Editor's Note: *The Visualization Toolkit* was reviewed in "Programmer's Bookshelf," *DDJ*, June 1997.]

In our first example, we'll decimate a CAD part that consists of 11,036 triangles. Listing One (listings begin on page 116) is the Tcl script that creates the results in Figure 2. Note that vtk uses a dataflow approach, or series of filters, to process the data. The first filter is a stereo-lithography reader. (Stereo lithography is a 3-D triangle format for reading and generating 3-D plastic parts.) Then, the decimation filter processes the triangles. Finally a mapper is connected to an actor, which represents the object in the rendered scene. The decimator is able to reduce the number of triangles to 1876, a total reduction of 83 percent. Note that the surface is rendered flat shaded to exaggerate the effects of the decimation.

An instance of the *vtkDecimate* class in Listing One is inserted into the visualization pipeline immediately after the reader object. Three important attributes to this class are the *TargetReduction*, *ErrorIncrement*, and *MaximumIterations* instance variables. *TargetReduction* specifies the reduction level desired. The algorithm attempts to meet this level of reduction, but, in some cases, topology limitations or error limits prevent it. The *ErrorIncrement* ivar is the amount the allowable error is incremented each iteration of the algorithm. Finally, the *MaximumIterations* limits algorithm execution in case the requested reduction cannot be achieved.

In the next example, we'll decimate a range image derived from a 3-D Cyberware laser digitizer. The digitizer can generate meshes of 1 million triangles in 15–30 seconds. It also captures the color of the surface which is applied as a texture map. (The data shown is a scan of our friend Fran's face when he happened to wander into the lab one day.) Figure 3(a), the original mesh, consists of 52,260 triangles, while Figure 3(b), the final one, consists of 2562 triangles—a 95 percent reduction. Figure 3(c) shows the edges of the triangles. In the original

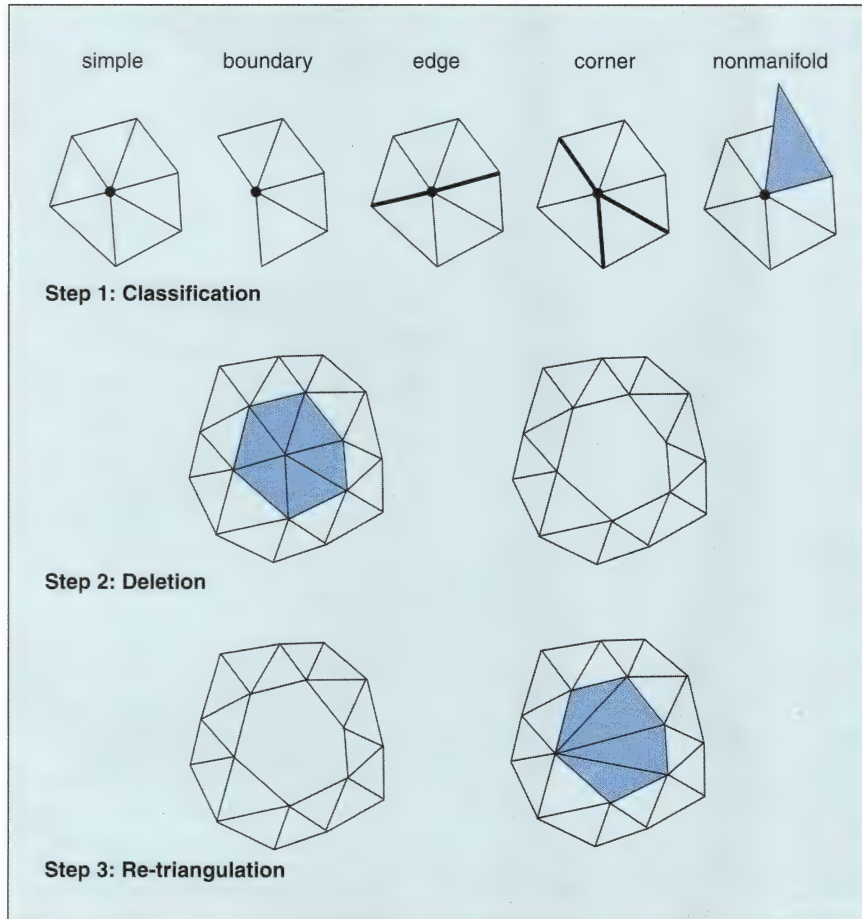


Figure 1: Overview of the decimation algorithm.

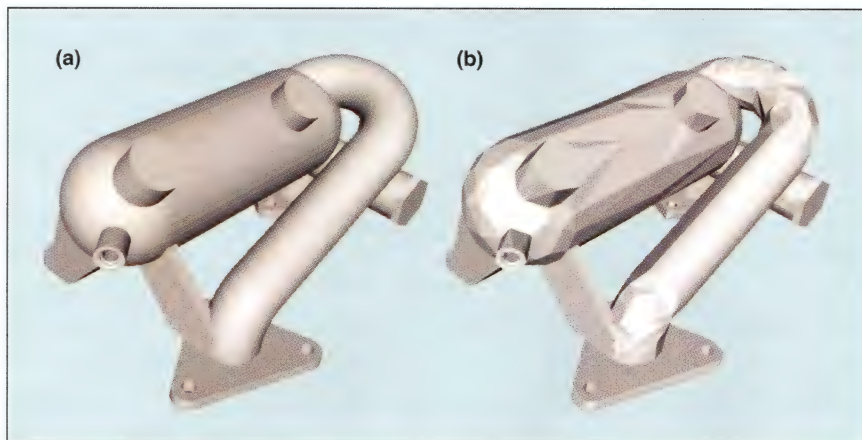
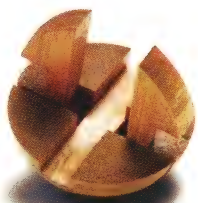


Figure 2: Decimation of a CAD part: (a) Original part shown flat shaded; (b) decimated part 83 percent reduction.



DB/EXPO NEW YORK



Today's data management and data warehousing technology is astounding, but falls short without a good system infrastructure and a high-performance network.



UNIX/WINDOWS NT® EXPO



Powerful operating systems and a seamless network are essential to any successful enterprise, but ineffective unless your data is well managed and accessible.



INTERNET/INTRANET EXPO



Both the Internet and intranets can revolutionize your business, but only if you combine them with smart data management and a solid operating system.



WHERE ALL THE PIECES COME TOGETHER.

The integration of all information technology into one powerful force—database management, applications, systems, networks, and now the Web and intranets—is poised to rule the future of business around the world.

Get the competitive edge you need to succeed: UNIX and Windows NT® EXPO, DB/EXPO, and Internet/Intranet Exposition all together in one powerful venue... IT FORUM '97.

CONFERENCE: SEPTEMBER 15-19, 1997 • EXPOSITION: SEPTEMBER 16-18, 1997 • JAVITS CONVENTION CENTER, NEW YORK CITY

HEAR 200 IT INDUSTRY SPEAKERS • SEE OVER 350 EXHIBITORS • JOIN OVER 40,000 IT PROFESSIONALS

Call 1-800-232-3976 today: Get your FREE 3-day EXPO pass • Receive a FREE feature-packed conference brochure—hot off the press • FREE IT FORUM updates via e-mail and much more.

7311

Name	Title	Company	Street	City	State	Zip
Mail Stop	Telephone	Fax	Email			

in Miller Freeman
Blenheim NDN

© Miller Freeman 1997

Or fill out and fax or mail this coupon to us right away. Fax: 1-415-966-8440
Mail: IT FORUM, 1975 W. El Camino Real, Suite 307, Mountain View, CA 94040
Or visit our Web site 24 hours a day: www.itforum.com

UNIX and Windows NT EXPO, DB/EXPO and Internet/Intranet Exposition are all registered trademarks of Miller Freeman. Windows NT is a registered trademark of Microsoft Corporation.



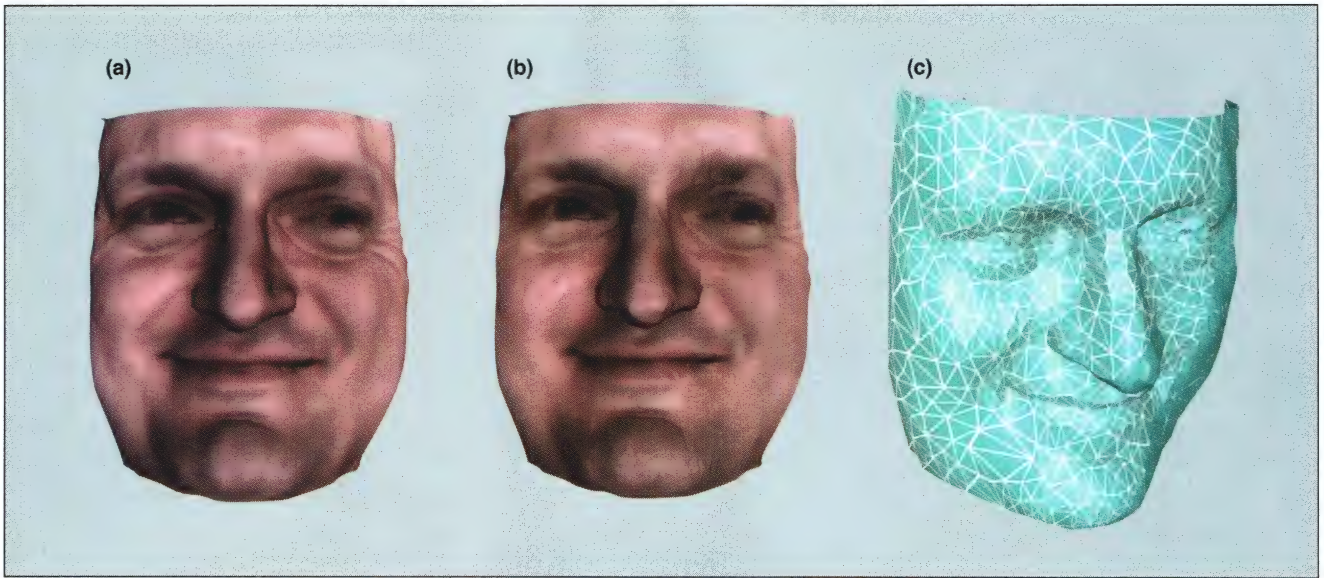


Figure 3: Decimation of CAD part: (a) original Cyberware data; (b) 95 percent reduction; (c) triangle edges.

(continued from page 110)

mesh, the edges were so dense that even a wireframe rendering of the surface appeared solid. Listing Two is a Tcl script that creates the images.

This example demonstrates the ability of surface texture to hide low-resolution geometry. Most interactive games make extensive use of texture to add detail to images. In fact, many 3-D graphics boards available on the PC are targeted toward low polygon count/textured surfaces. These boards do not perform well when the polygon count becomes large, since point transformation is often not accelerated.

In our final example, we'll parse a VRML 1.0 file, decimate the geometry, and then export the geometry using the *vtkVRMLExporter*. The parser was built using the QvLib parsing code developed by Gavin Bell and Paul Straus of SGI. The code currently reads in a VRML 1.0 file and creates an in-memory scene graph of the geometry. The system will deci-

mate any *IndexFaceSet* geometry nodes that exist in the file by placing a *vtkDecimate* object in the final rendering pipeline. The system then collects all geometry nodes, triangles and others, displays the resultant decimated geometry and outputs a VRML 2.0 file if specified on the command line.

The main function of the system is to translate the VRML scene graph into an object-oriented vtk scene. The translation encapsulates the scene graph by moving properties and transforms from hierarchical scene graph dependencies into encapsulated object pointers. This allows objects to implicitly represent themselves to external manipulators, an important feature for large-scale object systems that have many objects with multiple manipulators.

As these *IndexFaceSet* nodes are rendered in the object-oriented model, the *decimate* object takes the original triangles and outputs the decimated equivalent. Other shape nodes, such as Cones

and Cubes, are ignored since the browser is responsible for generating an optimal rendering solution for the specific shape.

Testing has shown that some VRML polygonal models can be reduced by 20–60 percent with little apparent visual change. Decimation can be used to automate the generation of multiple representations of an object for the VRML level of detail node. The result of one decimation is in Figure 4, which shows the original and decimated versions in SGI's Cosmo VRML viewer.

You can find the VRML decimator code at <http://www.rpi.edu/~citrit/>, plus additional examples and results. There are options to set parameters for the decimation algorithm via the command line. The code is available as a C++ distribution and requires vtk to be preinstalled. If you do use the VRML decimator, you will need a VRML browser to view the resulting models.

Conclusion

Triangle decimation algorithms play an important role in maintaining the responsiveness of 3-D graphics systems. By reducing data size, these algorithms minimize the amount of data to be stored on disk or in memory, transmitted across the network, and rendered by the graphics system. By compressing 3-D graphics data using triangle decimation, you can often squeeze out the last ounce of performance, which can make the difference between an engaging interactive system or one that annoys users.

DDJ

(Listings begin on page 116.)

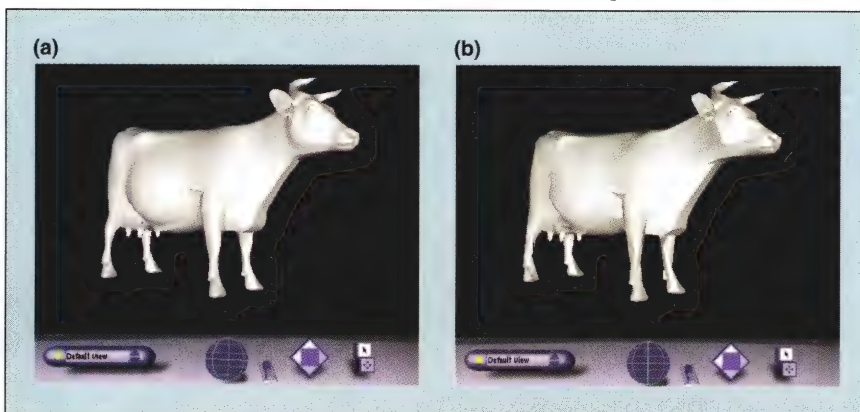


Figure 4: Decimation of VRML part: (a) original VRML data; (b) decimated VRML data. Total reduction is 70%.

In-Circuit Emulation

A powerful hardware tool for software debugging

Robert R. Collins



I launched a three-article discussion of Intel's System Management Mode (SMM) a few months ago with a comparison of the 80386's In-Circuit Emulator (ICE) support and System Management Mode (see *DDJ*, January 1997). Those columns provided more information about in-circuit emulation than I've seen in print in any other forum. Nevertheless, in-circuit emulation remains a mystery to most people. Engineers are spending scores of hours using arcane debugging techniques, simply because they don't know that a better, more powerful way exists to debug their toughest problems. The most common questions I am asked are: what is it? how much does it cost? what can it do? and what are the limitations? These are some of the questions I will attempt to answer in this column. I'll discuss the basics of in-circuit emulation, and the history of Intel's in-circuit emulator designs. In the coming months, I will discuss the microprocessor in more detail, as well as how it is designed with in-circuit emulation in mind. Finally, I will conclude the series by demonstrating some of the debugging techniques I've developed for use with in-circuit emulators.

My Own ICE History

My first experience with in-circuit emulation came as a junior-level BIOS engineer at Tandon Computers. I believed that I could solve any debugging problem using Microsoft Codeview. To me, Codeview represented the pinnacle of source-level debuggers. It let you set breakpoints and step through code execution in source or assembly-language mode. Breakpoints are set when Codeview surreptitiously inserts an INT-3 (breakpoint interrupt) into the code stream. The INT-3 insertion is hidden from the user when viewing in assembly mode, but the opcode can be detected when viewing raw memory. When

the breakpoint occurs, Codeview regains control and the user can view and modify the CPU registers, memory contents, and other program variables. This technique works fine when your program is in volatile memory (RAM). However, it never works when the program resides in read-only memory (ROM)—like the ROM BIOS. Obviously, Codeview cannot insert an INT-3 into the code stream of a ROM BIOS, because the ROM cannot be written. This is where the in-circuit emulator begins to help.

I was forced to start using an in-circuit emulator when I was given a problem that occurred in the ROM BIOS before the computer booted the operating system. Codeview and other debuggers can't run without first booting an operating system. Therefore, I knew that I couldn't use Codeview to help me debug this problem. My only alternative was to learn how to use the ICE. I cringed because I was scared to death of this monstrous ICE. In reality, I was afraid of learning something new. In fact, it didn't take long to learn how to use it; I actually liked it. The ICE was so powerful and versatile that I never looked back.

That was eight years ago. Today, I use an ICE for almost all my debugging. In many cases, a software-based debugger is still more productive than an ICE. However, over the years, I have honed my ICE debugging skills. Today, in those cases where a software debugger may have a productivity edge, my ICE skills often make up for the difference, giving me virtual parity in either environment.

In-Circuit Emulators versus Source-Level Debuggers

Modern in-circuit emulators share many attributes with software-based debuggers. Both support source-level debugging, software breakpoints, and memory watch points. You may reference or modify variables by name, or disassemble modules and subroutines by name and line number. Both allow you to view or modify raw

memory. Both give the ability to view and modify source code at the assembly-language level.

Even though ICEs have many of the same capabilities as software debuggers, they aren't without their limitations. Some ICEs support source-level debugging by loading executable programs (.EXE files) directly into memory, albeit without source-level debugging information (like line numbers and symbol tables). Even though Microsoft Codeview is the de facto standard for software debuggers, Intel never supported the loading of an executable file that was compiled with Codeview debugging extensions. Instead, Intel's ICEs only supported source-level debugging by loading object modules (.OMF files) directly into memory. The Intel OMF files were a pseudoproprietary format—clearly not an industry standard. If you wanted to convert your .EXE files (with Codeview information) into .OMF files (with Intel debug information), you needed an \$800 software package called "Link & Locate 386 Extended Edition," from Systems & Software (<http://www.ssi.com/>). Thankfully, Intel has exited the ICE business, allowing other ICE vendors to depart from their (backward) lead. Today, ICE manufacturers are starting to support source-level debugging in standard Microsoft (.EXE) formats. As far as I'm concerned, this limitation (along with the hefty price tag) is the worst limitation of using in-circuit emulators for every day debugging problems.

In all other ways, an in-circuit emulator is much more powerful than any software-based debugger. Breakpoints may be set anywhere in a program, regardless of the memory type (RAM or ROM). Breakpoints can also be set on certain bus events that cannot be detected by any source-level debugger. Consider the following breakpoint events, and ponder using a standard debugger to do the same thing:

- Break upon any memory access (*read*, *write*, or *either*) at a specific address,

Robert is an independent consultant on the x86 architecture. He can be reached at rcollins@x86.org.

with specific data.

- Break upon any memory access at a partial don't-care address, like 00124XXX with partial don't-care data such as XX2345XX.
- Break upon data access (memory or I/O, but not code fetches) to a specific address, or an address and data with don't-care fields.
- Break upon any byte, word, or double-word memory or I/O access at any address.
- Break upon any microprocessor special cycle, such as INT-Acknowledge, FLUSH, or a CACHE WRITEBACK.
- Break upon an interrupt acknowledge cycle for a specific interrupt, or any arbitrary interrupt.
- Break upon bus inactivity or ADS hold violations.
- Break upon code fetches with specific data patterns.
- Even break upon certain illegal bus cycle types.

The ICE has the ability to halt emulation on any of these events, and many more that don't readily come to mind. These capabilities are all accomplished with hardware assistance, which doesn't exist in software-based debuggers. This extra hardware assistance gives the ICE its powerful capabilities, but also gives it a hefty price tag. The traditional hardware-assisted ICE costs between \$35,000 and \$50,000. Other scaled-down versions, called In-circuit Test Probes (ITPs) or In-Circuit Debuggers (ICDs), lack the hardware assistance, but cost under \$10,000. (I will discuss the differences between ICEs and ITPs in my next column.)

ICEs also support debugging symmetric multiprocessor (SMP) environments. The ICEs have the ability to chain together and signal each other when one processor receives a breakpoint event. This ability gives the ICE the clear advantage in debugging any SMP problems.

In truth, the term "emulator" is actually a misnomer. The ICE is not emulating anything—it's a real CPU that is plugged directly into the CPU socket. The ICE just happens to have a hundred-wire umbilical cord attached to it. For the most part, the ICE runs at the full CPU speed. Therefore, those hard-to-find race conditions between software and hardware will still show up using an ICE.

ICE Evolution: Intel's Offerings

The first ICE I used was a dedicated in-circuit emulator computer from Intel. I called this "the big blue box" type of ICE. The entire ICE was housed in a self-contained computer, which included a monitor and keyboard. The ICE plugged into the CPU socket, and targeted a 6-MHz

80286 motherboard. Execution breakpoints were detected by snooping code prefetches on the microprocessor data bus. When a code prefetch matched your specification, the ICE would halt. Then, you could examine or modify memory, CPU registers, memory, and I/O ports.

Later, Intel introduced the I2ICE, a departure from the dedicated ICE machine approach. Users needed to provide a host computer, which typically needed to be an

In-circuit emulators are much more powerful than any software-based debugger

80286-class machine with some amount of expanded memory. (Yes, that's right...LIM-style expanded memory.) If your computer didn't have expanded memory, Intel would sell you an above-board unit to serve this purpose. The host computer attached to the I2ICE hardware via a standard serial connection—communicating at the lightning speed of 9600 baud. The I2ICE hardware was in a box approximately 30x24x10 inches in size (if memory serves me correctly). The most noticeable attribute about the I2ICE was its noise—the hardware had a fan on it that was very loud, making it hard to work in close proximity to this unit.

The I2ICE hardware had a limited set of changeable CPU modules. I recall having an 8086/8088, and an 80286 CPU module for our I2ICE. I also seem to recall that an 80186/80188 module was also available. Emanating from one end of the I2ICE hardware was a lead-shielded cable (the CPU umbilical cord), while at the other end was a small hardware module that plugged into the CPU socket. The real CPU laid on top of this small module (in close electrical proximity to the motherboard). Adapters were available to switch between CPU form factors—for example PLCC and PGA versions of the 80286.

With the next evolutionary change, the huge (and loud) I2ICE hardware was gone. In its place was a small, streamlined box that measured approximately 15x12x2 inches. This ICE was commonly called the "hot-plate" or "boiler-plate" ICE because

of its appearance. The host computer plugged into this small box, which in turn plugged into a small module that housed the CPU. Like the I2ICE, this ICE targeted multiple microprocessor environments. Modules were available for the 80286, and the newly released 16-MHz 80386.

About this time, Intel dabbled in UNIX-based ICEs. At one point, it shipped an ICE that was UNIX-software based. Instead of requiring a host computer, a VT-100 terminal was needed. Once the hardware booted UNIX, the ICE host autostarted itself. I'm only aware of an 80386 version of this ICE, though I wouldn't be surprised if other microprocessor targets were available. My most remarkable ICE story (the discovery of five or six undocumented opcodes) involved using this orphan ICE environment. This UNIX-based ICE disassembled all of the undocumented opcodes on the 80386 microprocessor. This inadvertent mistake gave me all of Intel's internal names for these undocumented opcodes. After I reverse-engineered the functions of these opcodes, I published my findings at <http://www.x86.org/secrets/opcodes>. This publication caused some concern within Intel's legal department that I had stolen this information. I was more than happy to tell their lawyer that my discovery was the result of Intel's sloppiness.

As microprocessors got faster and faster, Intel was forced to keep redesigning its ICE hardware. The next evolutionary change occurred when Intel developed the "Cobra" ICE chassis. The Cobra chassis was small, measuring approximately 12x10x3 inches in size. It was quiet and designed to target multiple microprocessor environments. The Cobra chassis was a modular design and had boards that could plug in and out. When a new CPU was designed, you simply bought the two plug-in boards, and saved over \$20,000 in hardware costs. The Cobra chassis supported the 80386 DX, 80386 SX, and initial offerings of the 80486 DX, 80486 SX, and 80487 SX. An optional high-resolution time-tag board could be purchased for an additional \$1500. The time-tag board provided cycle-accurate time-stamp information to the ICE bus cycle traces. The option also required a math coprocessor (80387) be plugged into the host computer. With a whopping 38,000 baud communications link, the Cobra was also much faster at communicating with the host computer. This higher baud rate substantially helped transfer microprocessor traces from the Cobra chassis to the host computer.

The Cobra chassis also supported functions for multiprocessor environments. Each Cobra chassis had two "SYNC" outputs and two "SYNC" inputs. These SYNC signals provided the means to start, stop, and trig-

ger multiple ICEs in a multiprocessor environment. These signals could also be used to send and receive signals to and from external devices. I often used these signals to trigger a logic analyzer or SCSI BUS analyzer when the CPU detected a breakpoint. Conversely, I used the logic analyzer and SCSI BUS analyzers to trigger breaks to the ICE using these signals.

Before Intel came out with the Cobra version of the 80486 ICE, they vowed that it couldn't be done. Intel's sales force told me that they couldn't design an ICE that ran at such a fast CPU speed (33 MHz). Intel promised us developers that the 80386 ICE was the last of the standard, hardware-assisted ICEs. Instead, Intel introduced the 80486 In-Circuit Debugger (ICD). The ICD didn't support any of the fancy hardware breakpoint features of a full-blown ICE. The ICD breakpoint ability was limited to breakpoint events that were supported by the microprocessor itself. This included any debug register breakpoint, trap flag breakpoint, INT-1, task switches into a task with the T-bit set, writing to the debug registers when DR7.GD=1, and the undocumented ICEBP instruction (see <http://www.x86.org/secrets/opcodes/ICEBP.html> for a description of the ICEBP instruction). Without the hardware assistance, the ICD didn't have an instruction trace. Even so, once a break-

point occurred, the ICD could disassemble assembly-language instructions, read and write memory, read and write I/O ports, and view all of the microprocessor registers—including the undocumented segment descriptor cache registers (see <http://www.x86.org/articles/loadall/#SB1> for a description of the segment descriptor cache registers). Even without hardware-assisted breakpoints or an instruction trace, the ICD was a very powerful debugging tool. As the first in-circuit debugging tool that cost under \$10,000, the ICD was also a very cost-effective debugging solution.

The final evolutionary change occurred in the same time frame that Intel abandoned designing and selling in-circuit emulators. Even though Intel abandoned the ICE market, they designed one last emulation device to coincide with this final evolutionary change in microprocessor design. Intel had abandoned their current method of supporting in-circuit emulation functions within the CPU (available from the 80286 to 80486), and adopted an entirely new approach (available in the Pentium and Pentium Pro processors). This new approach was called a "debug port," and required an entirely new emulator design. The new emulator design was called an "In-circuit Test Probe" (ITP). Like the ICD, the ITP didn't have any hardware assistance. Without hard-

ware assistance, its debugging capabilities were identical to the ICD. The ITP was also a very cost-effective debugging tool, still costing less than \$10,000. The ITP solution was both evolutionary and revolutionary. The ITP approach to in-circuit emulation is by far my favorite of Intel's designs.

Conclusion

I have barely touched the surface of discussing in-circuit emulation here. I've been using ICEs for eight years, and have a good understanding of how they work. I have watched Intel's evolutionary changes to ICEs. During that time, ICEs have gone from dedicated hardware machines to a host-target environment. I've seen monstrous ICE boxes and debuggers that are small enough to fit in your pocket. In my next column, I'll discuss the changes that took place in the microprocessors themselves. In-circuit emulation wasn't always provided by external hardware; considerable ICE support was designed into the CPU. I'll take a look at these changes and discuss their strengths and weaknesses. In another future column, I'll discuss the Pentium Probe Mode—ICE support built right into the Pentium Processor.

DDJ

Buried under a mountain of programming requests?

Track All Bugs, Requests, Fixes, Support Calls and more...

SX Tracker 1.1 is here to help!!!

SX Tracker allows software developers to manage and resolve problems and programming requests that threaten software quality and implementation schedules. It can notify users of changes made to their records via cc:mail, MS Mail and internet e:mail. It allows you to fully customize forms, reports and queries to fit your needs. It's easy to use and install. Want your customer to access their records over the web? No problem with SX Tracker's Web Interface. Try it today for free and you won't regret it!

Free evaluation copy on the web!

StormX Software, Inc.

www.stormx.com

Phone: (402) 553-6671

Fax: (402) 551 7699

7914 West Dodge Road #434

Omaha, NE 68144

CIRCLE NO. 104 ON READER SERVICE CARD

Putting You In Control...

SftTree Tree Control

Printer / Files	Description	Ready
3 Xerox DocuTech 135 (A)	1st Floor, Printer Ro...	☒ Ready
5 HP LaserJet III PS	2nd Floor, Room 228	☒ Ready
6 TREECTL.CPP	Sample Program	☒ Printing
7 TREEEX.RC	Resource Script	☒ Ready
8 BONUS.DOC	Additional document...	☒ Ready
9 HP LaserJet 4	2nd Floor, Room 206	☒ Printer Down
10 URGENT.DOC	Most urgent output	☐ Not ready

This powerful, easy to use tree control is available as a DLL or ActiveX (OLE) control.

From simple listbox with bitmaps to multi-line, multi-column, hierarchical data display, SftTree delivers! Single and multiple selection, even drag & drop are built-in. This 16-bit and 32-bit ActiveX (OLE) control supports most popular environments including VB, Delphi, Access, Visual FoxPro, etc. The DLL based control supports 16-bit and 32-bit applications, C, Visual C++, Borland C++, and includes C++ classes for MFC and OWL.

Support
C++
Information
Features

SftTabs Tab Control

This DLL-based custom control supports C, C++ (MFC, OWL), Windows 3.1, Windows NT and Windows 95 with over 50 different tab styles!

Add tab controls to your 16-bit and 32-bit applications with this easy to use tab control!

Get your free demo today!

Visit our web site for more information and to download our demos.

www.softelvdm.com

11 Michigan Ave
Wharton, NJ 07885

(201) 366-9618

FAX (201) 366-3984

softel vdm
NOT THE INDUSTRY STANDARD

JAVA

Listing One

```
currencyCheckbox.setLabel(buttonLabelBundle.getString("CurrencyLabel"));
standardCheckbox.setLabel(buttonLabelBundle.getString("StandardLabel"));
```

Listing Two

```
CurrencyLabel=Currency
StandardLabel=Standard
FlagImage=us_flag.gif
```

Listing Three

```
CurrencyLabel=W\u00e4hrung
StandardLabel=Standard
FlagImage=de_flag.gif
```

Listing Four

```
String imageName = buttonLabelBundle.getString("FlagImage");
java.net.URL imageURL = getClass().getResource(imageName);
image = getImage(imageURL);
```

Listing Five

```
if (standardCheckbox.getState())
    format = java.text.NumberFormat.getInstance(locale);
else
    format = java.text.NumberFormat.getCurrencyInstance(locale);
setDisplayValue();
```

Listing Six

```
<html>
<applet code="Calculator.class" width=325 height=250>
</applet>
</html>
```

ALGORITHM ALLEY

Listing One

```
catch (load vtkc1)
# get the interactor ui
source vtkInt.tcl
```

On November 2, 1997, **Java** programmers
from around the world will gather in Colorado's
Rocky Mountains and change the world.

Where will you be?

Finally, a **Java™** conference devoted
exclusively to providing you with the latest
and most comprehensive technical knowledge
available on the full range of **Java** software

The **Java** conference for professional
programmers.

"Just what I needed!"
"Just when I needed it!"
"Straight talk... straight from the best!"

**Colorado
Software Summit**

**Programming Conference
Featuring Java**

November 2-7, 1997
Keystone Resort, CO

To register or for more information:
Voice: 800.481.3389 USA & Canada

719.481.3389 International
Fax: 719.481.8069

For complete details:
<http://www.SoftwareSummit.com>

Java is a trademark of Sun Microsystems Inc.

Colorado
Software Summit '97

It just might be the
most important week
of your career!

Java Java Java Java Java Java Java
Java Java Java Java Java Java Java
Java Java Java Java Java Java Java

CIRCLE NO. 366 ON READER SERVICE CARD

```
# Read in CAD part
#
vtkSTLReader sr
sr SetFileName ".../data/42400-IDGH.stl"
vtkDecimate deci
deci SetInput [sr GetOutput]
deci SetTargetReduction 0.9
deci SetErrorIncrement 0.002
deci SetMaximumIterations 8
deci DebugOn
vtkPolyMapper stlMapper
stlMapper SetInput [deci GetOutput]
vtkLODActor stlActor
stlActor SetMapper stlMapper

# Create graphics stuff
vtkRenderMaster rm

# Now create the RenderWindow, Renderer and both Actors
set renWin [rm MakeRenderWindow]
set ren1 [$renWin MakeRenderer]
set iren [$renWin MakeRenderWindowInteractor]
# Add the actors to the renderer, set the background and size
$ren1 AddActors stlActor
$ren1 SetBackground 0.1 0.2 0.4
$renWin SetSize 450 450

vtkCamera cam
cam SetClippingRange 2.21973 110.986
cam SetFocalPoint 128.766 86.0079 224.742
cam SetPosition 133.866 68.8423 211.626
cam SetViewAngle 30
cam SetViewPlaneNormal 0.2298 -0.773329 -0.590893
cam SetViewUp -0.0185622 0.603548 -0.797111
$ren1 SetActiveCamera cam

# render the image
$iren SetUserMethod [wm deiconify .vtkInteract]
$iren Initialize

# prevent the tk window from showing up then start the event loop
wm withdraw .
```

Listing Two

```
catch (load vtkc1)

# this is a tcl version to decimate fran's face
# get the interactor gui
source vtkInt.tcl

# create visualization pipeline
# create decimated model with surface normals
vtkCyberReader cyber
cyber SetFileName ".../data/fran.cut"
vtkDecimate deci
deci SetInput [cyber GetOutput]
deci SetTargetReduction 0.95

deci SetErrorIncrement 0.002
deci SetMaximumIterations 8
deci DebugOn
vtkPolyNormals normals
normals SetInput [deci GetOutput]

# ingest image and apply as texture map
vtkPNMReader pnm
pnm SetFileName fran_cut.ppm
vtkTexture aTexture
aTexture SetInput [pnm GetOutput]
aTexture InterpolateOn

# map to graphics system
vtkPolyMapper cyberMapper
cyberMapper SetInput [normals GetOutput]
vtkActor cyberActor
cyberActor SetMapper cyberMapper
cyberActor SetTexture aTexture

# Create graphics objects
vtkRenderMaster rm

# Create the RenderWindow, Renderer and both Actors
set renWin [rm MakeRenderWindow]
set ren1 [$renWin MakeRenderer]
set iren [$renWin MakeRenderWindowInteractor]

# Add the actors to the renderer, set the background and size
$ren1 AddActors cyberActor
$ren1 SetBackground 1 1 1
$renWin SetSize 450 450

# Define method to invoke when "u" is pressed in graphics window
$iren SetUserMethod [wm deiconify .vtkInteract]

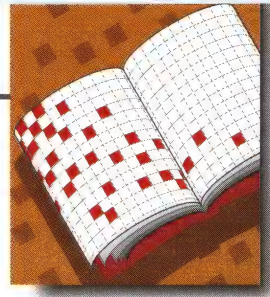
# Define nice view
vtkCamera cam1
cam1 SetClippingRange 0.0475572 2.37786
cam1 SetFocalPoint 0.052665 -0.129454 -0.0573973
cam1 SetPosition 0.327637 -0.116299 -0.256418
cam1 CalcViewPlaneNormal
cam1 SetViewUp -0.0225386 0.999137 0.034901
$ren1 SetActiveCamera cam1

$iren Initialize

# prevent the tk window from showing up then start the event loop
wm withdraw .
```

DDJ

Dr. Dobb's Journal, July 1997



Tools, Languages, and Interacting with Machines

Gregory V. Wilson

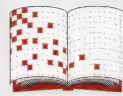
Most authors write because they hope to strike it rich, or because it's part of their job, or because they have something they're itching to tell you about. Jonathan Rosenberg's *How Debuggers Work* falls into the third of these categories. After helping to develop the tools that made MasPar's MP machines easier to program than most parallel supercomputers, Rosenberg moved to Borland, where he worked on debugging tools. Having read the blurb on the back of this book, I hoped it would be for debuggers what Tanenbaum's *Operating Systems: Design and Implementation* was for its subject: a practitioner's guide to what matters when you actually have to get something built.

Unfortunately, *How Debuggers Work* falls short. Rosenberg covers all the right topics—hardware and OS support, breakpoints, context and data inspection, multithreading, GUIs, and how to handle optimized code—but each time I thought he was about to get into the nuts and bolts, the section or chapter ended.

For example, after pointing out that the PowerPC can normally execute instructions out of order, but always executes them in order when debugging, he says, "This is something for debugger developers and users of the PowerPC to watch closely." I'm grateful for the warning, but I would have been more grateful if he'd told me what I could do about it. Similarly, after pointing out that it can be difficult to set breakpoints on single-line *for* loops and outlining a simple-minded solution, Rosenberg says, "This solution is not foolproof..." and then he stops.

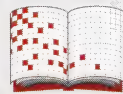
How Debuggers Work could also have benefited from more careful editing. The list of Java debugging classes, for instance,

Greg is the author of *Practical Parallel Programming* (MIT Press, 1995), and coeditor with Paul Lu of *Parallel Programming Using C++* (MIT Press, 1996). Greg can be reached at gvwilson@interlog.com.



How Debuggers Work

Jonathan B. Rosenberg
John Wiley & Sons, 1996
265 pp., \$34.95
ISBN 0-471-14966-7



Programming Language Choice

Mark Woodman
International Thomson
Computer Press, 1996
384 pp., \$42.95
ISBN 1-85032-186-5



The Media Equation

Byron Reeves and Clifford Nass
Cambridge University Press/
CSLI Publications, 1996
305 pp., \$27.95
ISBN 1-57586-052-X

contains the meaningless "*public class extends RemoteValue*" seven times, a mistake that is repeated in the tables later. Someone should also tell the author that the Heisenberg principle doesn't actually say that "[a] debugger must intrude on the debuggee in a minimal way."

Having been so critical, I should now say that I think this book has a lot of potential. None of the undergraduate computer-science programs I'm familiar with teach students how to build tools for debugging, profiling, dependency management, or version control. As a result, most undergraduates enter the workforce either not knowing how to use such tools, or believing that they aren't really important. (After all, if they were as important as parsers, we'd have done four homework assignments on them

as well, wouldn't we?) If Rosenberg were to turn this book into a practical guide of the sort that exists for operating systems and compiler construction, he would do computing a great service.

Programming Language Choice

The second book on my list, *Programming Language Choice*, was an even greater disappointment. Sooner or later, everyone in computing gets embroiled in a debate over the relative merits of different languages. A few years ago, after sitting through yet another presentation in which yet another parallel programming language was called "intuitive, powerful, and flexible," I decided to look through the literature for experimental comparisons of the usability of different programming languages.

I was surprised to discover that few have ever done such experiments. Think about that for a second: Almost everyone believes that some languages (Smalltalk, C++, Perl, Scheme) are "better" than others (Fortran and Cobol being the usual targets for scorn), but I was only able to find one group that had ever tried to quantify the differences ("An Experiment to Measure the Usability of Parallel Programming Systems," by Duane Szafron and Jonathan Schaeffer in *Concurrency: Practice and Experience*, 8(2), 1996). So much for the "science" in computer science.

When I mentioned this to colleagues, they were less surprised than I had been. "There's so much variability among programmers that no study could be meaningful," was a common reaction, while another was, "The needs of different domains are so different..." However, pharmacologists manage to study the effectiveness of drugs, despite variations in sensitivity of up to 10 million to 1, and they don't seem bothered by having to specify an application area before starting a study. Even in computer science, we measure the usability of GUIs. Surely, I thought, something similar could be done for the one tool that every programmer uses?

JAVA

ACTIVEX

C++

500 TOOLS

300 EXHIBITORS

200 CLASSES

EXPERT SPEAKERS

15,000 DEVELOPERS

PEER NETWORKING

FREE SOFTWARE

FOR THE SD CATALOG:

800.441.8826

415.905.2702

SD97@MFI.COM

It
sets
your
brain
on
fire



SOFTWARE DEVELOPMENT '97

CONFERENCE & EXHIBITION

SEPT. 29-OCT. 3, 1997

WASHINGTON, DC

You arrive at SD '97. You

learn about hundreds of

new tools and technologies.

Synapses commence firing.

Fresh techniques for

building killer apps. Senses

are sharp. Innovative ways

to combine object,

C/S and web technologies.

Brain begins to sweat.

Smart strategies for teams.

Powerful enterprise systems

Fast, useful JavaBean

components. Damn, it's

intense in here.



un Miller Freeman
A United News & Media company

WWW.SD97.COM

TRAVELING AT THE SPEED OF DEVELOPMENT

(continued from page 117)

When I came across this book, I hoped that it might be a step toward that goal. No such luck. With a few exceptions, this book might as well be titled *Programming Language Advocacy*. Most contributors simply describe their favorite language, then claim that it is well suited to undergraduate teaching, without offering any evidence stronger than their personal preferences. At its worst, this advocacy degenerates into hype. Do you believe, for example, that "Almost all Smalltalk projects succeed," or, "While other languages were being deliberately shaped to encourage particular design approaches, Lisp...could mold itself around any approaches that seemed appropriate"?

Matters aren't helped by the fact that the conference that gave rise to this book was held in September 1993. As a result, there is no mention of Java, whose ubiquity, safety, and object orientation make it a very strong candidate for undergraduate education, while Ada 95 is spoken of in the future tense, and defunct research languages like Cool and Omega are given as much space as Smalltalk and C++.

Programming Language Choice does have two redeeming features. The first is Marian Petre's discussion of programming paradigms and cultures, which is worth

reading carefully. She points out that most programmers unconsciously shift the paradigms of the language they happen to be working in. Similarly, in the absence of a clearly defined model of what the computer is doing, both expert and novice programmers will borrow from whatever models they have and construct a "just-so" story

*Sooner or later,
everyone gets
embroiled in a
debate over the
relative merits of
different languages*

to explain the behavior they observe. I also enjoyed the chapter by Lee and Stroud on C++, primarily because it laid out, in detail, the criteria to use in choosing a teach-

ing language. These discussions make the book worth borrowing, but you'll probably want to spend your money elsewhere.

The Media Equation

The same is true of *The Media Equation*, even though it is a much better book. Reeves and Nass both work at Stanford University, where they study how people react to and interact with computers, television, and other media. They have discovered that even when people know they're dealing with a machine, they treat it as if it were a social being—a person. Similarly, people who've been flattered by a computer program will think that they've done better work with it, even if they know that it's just a program and that its flattery is preprogrammed.

This may sound a bit trite, but the authors' initial detailed description of their experiments and results is fascinating. However, I was less fascinated by their third and fourth detailed descriptions, and by Chapter 6, I was really hoping to see a sentence starting, "We can summarize our other work as follows..." Topics discussed in the book's later chapters aren't any less interesting, but after 200 pages, I'm more interested in conclusions than experimental design.

DDJ

NT FILE SYSTEMS and DEVICE DRIVERS CLASSES

Learn from the experts the "big guys" call.

CLASSES:

- 1 DEVELOPING FILE SYSTEMS for WINDOWS NT – 5 day course
- 2 WINDOWS NT KERNEL MODE DEVICE DRIVERS – 3 day course

START DATES:

	1	2
Boston	5/5/97	5/5/97
Tel Aviv, Israel*	5/19/97	5/19/97
Munich, Germany*	5/26/97	5/26/97
Denver	---	6/2/97
Taipei, Taiwan*	6/16/97	6/16/97
Palo Alto	7/14/97	7/14/97

*Prices for non-U.S. classes higher. Contact OSR for details.

1 - 8 8 8 - 6 7 7 - 4 2 6 4

Call for details and fees. Or visit OSR's NT World at www.osr.com.

O S R · Open Systems Resources, Inc.

Customized Development, Consulting and Training
105 Rte. 101A, Suite 19 • Amherst, NH 03031 USA
voice: +1 (603) 595-6500 • fax: +1 (603) 595-6503
e-mail: info@osr.com • web: www.osr.com

EXCEED YOUR EXPECTATIONS

CIRCLE NO. 339 ON READER SERVICE CARD

Protect your business. Secure your data and software.



Whether it's the illegal duplication of software you developed or the unauthorized access of proprietary data, *piracy steals your revenue*. Secure your software, data and profits with UniKey®, the piracy solution from the industry's most experienced provider.

UniKey Protects Better

- cross platform flexibility
- remote activation & upgrades
- no access to source code required
- parallel or serial port option
- compatibility with DOS, Windows, 95, NT, UNIX and NetWare



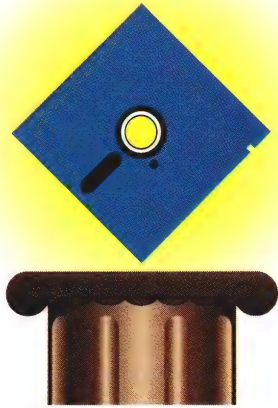
Respond now for a **FREE**
Information Kit
& **Test Drive** disk.

**SOFTWARE
SECURITY**

Call 800-841-1316 or
visit www.softsec.com

Tel: (203) 656-3000 ■ Fax: (203) 656-3932 ■ Email: softsec@softsec.com
6 Thorndal Circle, Darien, CT 06820 ©1997 Software Security, Inc. All rights reserved.

CIRCLE NO. 360 ON READER SERVICE CARD



Newfire has unveiled Catalyst, an authoring tool for creating 3-D games for the World Wide Web. Catalyst is designed as an extensible base platform that can be extended through plug-ins. The Catalyst base platform supports/provides scene composition, an interactivity editor, real-time playback and analysis, performance optimizer, and project manager. A Catalyst SDK will be available shortly. The Catalyst base product is priced at \$1995.00 for a single-user license. Newfire's plug-in modules are priced between \$250.00 and \$1000.00 each.

Newfire Inc.
12901 Saratoga Ave., Suite 4
Saratoga, CA 95070
408-996-3100
<http://www.newfire.com/>

Pro\Sim has announced Version 2 of its PinQuik (PQ) Toolkit for system administrators and developers. PQ is a user-interface development package, which can simultaneously execute in graphical and text modes. PQ is delivered with sample panels, bar menus, and forms that can be used as examples for users to start developing their own applications. PQ has several UNIX System Administration Utilities built in, including disk utilities, administration tools, and others. PQ is implemented in Tcl/Tk, and a copy of Tcl/Tk is distributed with PQ.

Pro\Sim Corp.
14201 Memorial Dr., Suite 400
Houston, TX 77079
800-489-3592
<http://www.prosim.com/>

Headspace's Beatnik is a web-based software system for the creation and playback of interactive music. The Beatnik system consists of the Beatnik Editor and the Beatnik Plug-in. The Beatnik Editor, available for the Macintosh, can import music and sound in a variety of industry-standard file formats, and outputs Real Music For-

mat (RMF), which allows for MIDI music to be integrated with digital audio samples. The Beatnik Plug-in manages playback on web browsers. Beatnik supports a set of Java-callable functions that allow a web site to play music based on user events, such as a mouse-click or a mouse-over. These events can also trigger individual notes, sampled voices, or sound effects; start or stop music; and change tempo, volume, pitch, or mix. The Beatnik Plug-In and the Beatnik Editor are both freely available.

Headspace Inc.
217 South B St.
San Mateo, CA 94401
415-696-9400
<http://www.headspace.com/beatnik/>

TracePoint Technology announced HiProf, a hierarchical profiler available for the Windows Intel platform. HiProf's hierarchical profiling enables developers to conduct code analysis on the binaries of 32-bit C/C++ Windows applications. Hierarchical profiling shows each function's self-time and the number of calls taking place between all parent and child functions, and the time that the activity takes. This information is displayed in graphical pie charts, histograms, and lists. HiProf instruments and profiles compiled and linked EXEs or DLLs, using its patented Binary Code Instrumentation (BCI) technology. No source code or OBJ files are required with HiProf. HiProf also enables developers to selectively exclude DLLs or portions of their EXE file from profiling in order to drill down on specific portions of an application, or to evaluate third-party components. HiProf reports on multiple threads individually, and supports all applications compiled with Visual C++ 2.x and later. HiProf's suggested retail price is \$599.00.

TracePoint Technology Inc.
One Almaden Blvd., Suite 1100
San Jose, CA 95113
408-283-5350
<http://www.tracepoint.com/>

DFL Software has announced 32-bit ActiveX versions of its Light Lib Magic Menus component. This control allows developers to create menus with background images, bitmap menu items, and tool button palettes. The professional version of Light Lib Magic Menus is \$139.00, while the standard version is \$99.00.

DFL Software
55 Eglinton Ave., East, Suite 208
Toronto, ON
Canada M4P 1G8
416-487-2660
<http://www.dfl.com/>

ObjectSpace's Voyager is a platform for agent-enhanced distributed computing in Java. It supports both traditional and agent-enhanced distributed programming techniques to create sophisticated network applications for equipment ranging from desktop computers to embedded consumer devices. Voyager enables you to create agents that can roam a network and continue to execute as they move. Moving agents and other objects can exchange remote messages using regular Java message syntax. Objects can be constructed remotely using the regular Java construction syntax, static methods can be executed remotely, remote exceptions are automatically rethrown to the caller, and serializable objects can be passed and returned by value. Once serializable objects are created, you can move any of these objects to a new location, even while the object is receiving messages. Messages sent to the old location are automatically forwarded to the new location. The entire Voyager system, is 70 KB in a compressed jar file, and 150 KB uncompressed. Voyager is available free for commercial use at the ObjectSpace web site.

ObjectSpace Inc.
14881 Quorum Dr., Suite 400
Dallas, TX 75240
972-934-2496
<http://www.objectspace.com/>

WebRunner Toolkit from Taligent is a web-development toolkit consisting of Client Works, Server Works, and Bean Factory. WebRunner Client Works is a set of Java class libraries that form a framework for JavaBeans and Java applications. WebRunner Server Works is a class library that abstracts common server-side APIs (such as CGI and NSAPI) so that you can develop API-independent server-side web applications. C++ and Java versions are available. Bean Factory is a tool for creating, modifying, and testing JavaBean components.

Taligent Inc.
10355 N. De Anza Blvd.
Cupertino, CA 95014
408-255-2525
<http://www.taligent.com/>

SoftQuad International has introduced the SoftQuad AdaptAbility Toolkit, a suite of technologies designed to make the World Wide Web accessible to people with disabilities. SoftQuad AdaptAbility Technologies consists of four components: a Visual Dynamic Keyboard, a Screen Enhancer, an integrated screen reader, and customizable user profiles. The Visual Dynamic Keyboard is an on-screen keyboard that lets users enter text, select commands,

THE ADVERTISER INDEX

RS#	Advertiser Name	Page #	RS#	Advertiser Name	Page #
124	Abraxas Software, Inc.....	94	*	PowerBar	25
65	Accusoft	7	295	Powersoft.....	61
605	Adobe Systems.....	124	121	Premia	8-9
*	Al Steven's Cram Course CD-ROM.....	86	*	Product Debut	123-124
143	American Cybernetics	49	*	Programmer's Marketplace.....	125-127
356	American Power Conversion	33	128	Programmer's Paradise.....	14-15
152	Asymetrix.....	13	315	Pure Atria.....	39
*	AT&T	5	134	QNX	47
604	Automation Technology, Inc.....	124	*	Quinn-Curtis	11
282	Avocet Systems Inc	53	*	Raima	98
180	Blue Water Systems	78	603	Raleigh Group International, Inc	123
67	Borland International	C4	607	Red Hat Software.....	124
68	Bristol Technology	21	215	Rogue Wave Software.....	56-57
170	Bristol Technology	63	215	Rogue Wave Software	77
366	Colorado Software Summit	116	73	Samson Info Tech	43
324	Computer Management Group	44	203	Sandstone Technology	38
131	Connect Tech, Inc	75	105	Sequiter Software, Inc.....	C3
80	Data Techniques.....	22	76	Smartcode Software	87
151	Datalight.....	48	309	Snowbound Software.....	82
606	DC Micro Development	124	*	Softel VDM.....	115
348	Dinkumware	79	360	Software Security, Inc	119
125	Distinct Corporation	62	236	Springer-Verlag Publishing.....	71
*	Dr. Dobb's Alternative Languages CD-ROM.....	52	286	Stingray Software	90
*	Dr. Dobb's Books on Graphics Programming ..	64	104	StormX Software.....	115
*	Dr. Dobb's CD Release 5	103	157	Sub Systems, Inc	26
*	Dr. Dobb's Small C CD-ROM	108	362	Sun Microsystems.....	73
*	Dr. Dobb's Web Site	74	195	Symantec Corp.	104
*	Dr. Dobb's Encryption & Security Books	92	601	Syware.....	123
*	Dr. Dobb's CD-ROM Library	96-97	337	Technical Resource Connection	29
609	Dundas Software Ltd.....	124	611	Ted Gruber Software	125
*	Embedded Systems Conference	107	608	Tetradyne Software	124
84	FairCom	4	359	Tracepoint Technologies	67
77	Fidelity Investments	41	*	Vireo	42
*	Game Developer.....	100	361	Visigenic	78
*	Gimpel Software.....	27	600	WCSC.....	123
367	Green Hills Software Inc	58	602	WIBU	123
87	Greenleaf Software	88	*	World Wide Web Directory	127
349	Greenview Data, Inc	101			
*	IT Forum	111			
310	Kenonic Controls Limited	79			
288	KL Group, Inc.....	94			
90	Lead Technologies	80			
146	MDBS	83			
160	MicroEdge, Inc.....	37			
*	Microsoft.....	17			
*	Microsoft.....	89			
*	Microsoft.....	C2-1			
*	Microsoft Press	65			
354	MindQ Publishing.....	91			
610	Netwin Ltd	124			
342	NuMega Technologies.....	45			
172	ObjectSpace	31			
*	Object World Expo.....	85			
*	On Time Informatik GmbH.....	59			
339	Open Systems Resources, Inc.....	119			
232	Opus Software	75			
340	Pacific Softworks.....	51			
99	Phar Lap.....	54			
*	Poet Software.....	34-35			

The index on this page is provided as a service to our readers. The publisher does not assume any liability for errors or omissions.

*The advertiser prefers to be contacted by phone; consult ad.

**See individual ads for RS#.

Advertising Sales Offices

Paul Miller	National Accounts Manager	(415) 655-4183
Mid-Atlantic/Southeast		
Nicole Glazen	Accounts Manager	(617) 235-8568
California/Southwest		
Carrie Mills	Accounts Manager	(415) 655-4196
David Katch	Sales Associate	(415) 655-4301
Northwest/Mountain/Texas		
Aimee Einstein	Accounts Manager	(415) 655-4221
Michael Beasley	Sales Associate	(415) 655-4304
Northeast/Northcentral		
Brenner Fuller	Sales Manager	(603) 746-3057
Linda Guyette	Sales Associate	(603) 924-5971
Software Careers		
Lilly Gutierrez	National Accounts Manager	(415) 655-4193

(continued from page 120)

activate dialog box controls, and so on, without using a traditional keyboard and mouse. The Screen Enhancer is a screen-magnification technology that rerenders the desktop and/or the application in logical ways for better, more intuitive navigation and viewing. SoftQuad's integrated screen reader is both application aware and HTML aware. In addition to standard screen-reader elements, users are able to set up automatic routines for reviewing an HTML page when it appears. SoftQuad's Adaptable Technologies also include customizable user profiles, in which users can select display and control preferences. User profiles can be selected upon startup, so multiple individuals with varying preferences can use a single terminal.

SoftQuad Inc.

20 Eglinton Ave. West, 12th Floor
Toronto, ON
Canada M4R 1K8
416 544-9000

<http://www.softquad.com/>

Arial Software's Virtual Print Engine gives developers using Visual Basic, C/C++, FoxPro, Delphi, and other languages the ability to dynamically create and output documents by invoking any of Virtual Print Engine's 65 page-building function calls. Objects such as lines, polygons, bitmaps, text, and barcodes can be positioned, rotated, and scaled, and can incorporate external data at run time. Developers can give users print previewing. The Virtual Print Engine has no run-time fees or royalties, and a list price of \$398.00.

Arial Software

1225 NW Murray, Suite 101
Portland, OR 97229
503-646-4515

<http://www.arialsoftware.com/>

Snowbound Software's RasterNote Annotation Toolkit for Windows 3/95/NT lets you use circles, lines, and freehand shapes to annotate documents. RasterNote toolkit is available as a DLL for Windows 3.x, and OCX/ActiveX for Windows 95/NT, and costs \$995.00.

Snowbound also announced version 6.0 of its RasterMaster UNIX image libraries. New features include enhanced document imaging support for hot document archival, deskewing, despeckling, automatic aspect ratio correction, and GIF and PNG support. The RasterMaster toolkit for UNIX costs \$2495.00.

Snowbound Software Corp.

P.O. Box 520
Newton, MA 02159
617-630-9495

<http://www.snowbnd.com/>

RoseGUI from HexaTech is an ActiveX component for developing GUIs. Features include creating interactive scenes containing any objects, shapes, and behaviors of objects controlled by user bitmaps, simulating real-life objects such as airplanes and TVs, support for flying-mouse highlight, and scripting support. RoseGUI allows you to supply your own bitmaps to control the look and feel of your application.

HexaTech

725 Mariposa Ave., #107
Mountain View, CA 94041
415-254-0610

<http://www.hexatech.com/>

Interactive Network Technologies, has announced Version 1.1 of Carnac, a 2-D graphics toolkit. Two SDKs have been added to the toolkit: the Flavors SDK and Dynamic Scenes SDK. The Flavors SDK aids production of User Defined Primitives (complex shapes), while the Dynamic Scenes SDK assists production of code that generates dynamic scenes on the fly. Additionally, support for 1, 8, and 24-bit rotatable, sharable, and scalable images with full transparency control and a CGM output option have been added. Carnac is available for Windows 95/NT and most common UNIX implementations.

Interactive Network Technologies Inc.

2901 Wilcrest, Suite 300
Houston, TX 77042-6011
713-975-7434

<http://www.int.com/>

Preview Software announced TimeLOCK 3.0 Transaction Server and Client Builder, a client/server extension to Preview's trialware development kit. TimeLOCK Client Builder allows you to wrap your software with "try before you buy" and electronic commerce capabilities. The Client Builder supports multitier distribution as well as direct sales, and allows connections via the Internet, modems, voice, and fax. With the TimeLOCK Transaction Server, software companies can handle all aspects of online transactions royalty free. The Transaction Server is available for both Windows NT and Solaris, and a single-key license costs \$4995.00. A license for TimeLOCK Client Builder and one key module costs \$795.00.

Preview Software

345 California Ave., Suite 3
Palo Alto, CA 94306
415-326-2242

<http://www.previewsoft.com/>

EdgeworX, from the Antares Alliance Group, is an object-oriented web development and execution framework using Microsoft's Visual Basic for Applications

(VBA). EdgeworX uses a class library of ActiveX objects to represent the base components of a web application, including objects for database access. EdgeworX works on any Windows 95/NT web server that supports CGI, ISAPI, or NSAPI, and costs \$399.00.

Antares Alliance Group

17304 Preston Rd., Suite 1200
Dallas, TX 75252

972-447-5500

<http://www.aag.com/>

emWare has announced EMIT (Embedded Micro Interface Technology), a micro web server designed for embedded devices. EMIT requires 750 bytes of ROM and 30 bytes of RAM. It achieves its size by distributing part of the server functionality to the client, emManager. The client interface consists of emObjects, pre-programmed functions in the form of Java applets, ActiveX components, and others. A single-user EMIT SDK with ten embedded licenses costs \$1200.00.

emWare

1225 E. Fort Union Blvd., Suite 220
Midvale, UT 84047

801-256-3883

<http://www.emware.com/>

NetManage has announced Chameleon UNIXLink 97, which uses the X Consortium's Broadway standard to run X applications from within a web browser. Chameleon UNIXLink 97 uses LBX to handle data over low-bandwidth lines, and includes a web-based management tool. Chameleon UNIXLink97 sells for \$400.00.

NetManage Inc.

10725 North De Anza Blvd.
Cupertino, CA 95014
408-973-7171

<http://www.netmanage.com/>

Rational Software has announced new versions of SQA Suite 6.0 and RequisitePro 2.5, both of which have been integrated with the Rational Rose visual modeling tools. SQA Suite 6.0, a suite of automated client/server testing tools includes new capabilities for HTTP-based performance testing and a new Object Scripting capability. RequisitePro 2.5, a requirements management tool, adds integration with Microsoft Visual SourceSafe, in addition to Rational Rose. SQA Suite 6.0 starts at \$9995.00, while RequisitePro starts at \$1295.00.

Rational Software Corp.

2800 San Tomas Expwy.
Santa Clara, CA 95051

408-496-3600

<http://www.rational.com/>

DDJ

PRODUCT DEBUT

Supports
Windows 95 &
Windows NT

SERIAL I/O

The **ONLY** Serial Communication Libraries, DLLs, & Tools
You Will Ever Need For Windows 3.x, Windows 95,
Windows NT, and MS-DOS. Accept No Less Than The Best.

<http://www.wcsnet.com/home.htm>

COMM-DRV/LIB™

Professional Serial I/O Libraries & DLLs for
Windows 3.x, Windows 95, NT, & MS-DOS

- Supports ALL languages, tools, or applications that can call the Windows API.
- Complete source code to all libraries & DLLs.
- Same API for Windows 3.x, Windows 95, NT, & MS-DOS.
- High level Hayes compatible modem functions.
- X, Y, Zmodem file transfers on multiple ports.
- Asynchronous & timed callback to user functions on different serial communication events (modem signals, buffer counts, character reception, and more).
- Supports Visual C/C++, ANSI C/C++, QuickBasic, Visual Basic (MSDOS & Windows), Assembly, Access, Transmit data from user callback on any event.
- Extensive scanning of input character stream.
- Support most intelligent all dumb multipoint cards (Amnet, AST, Boca, Cyclades, Digiboard, GTEK, & many more).

MS-DOS Serial I/O TSRs & Utilities

- True DOS device driver that allows serial ports to be opened like files under MS-DOS & Windows.
- FOSSIL interface for all supported Supports all BBS, Mailers, etc. (PCBoard, Wildcat, Frontdoor, Doorway, Renegade, etc.).
- Provides Int14h, Int21h, & DAM interfaces.
- Compatible with Desview and Windows.
- Real time serial port monitoring to screen and disk.
- Mini-BBS for file transfer Spoolable file transfer engine.
- Supports most intelligent & non-intelligent multipoint cards.

COMM-DRV/VxD™

Ultra High Speed Serial I/O VxD For Windows

- Windows (95 & 3.x) DLLs with API calls into the 32bit VxD.
- MS-DOS high speed libraries for calling VxDs from a DOS box.
- All serial communication interrupts at ring 0 in 32 bit mode.
- Uses transmit & receive FIFOs (16550, 16550, 16650, etc.).
- Win32, Win16, Console, & MS-DOS apps active concurrently.
- Over 400K baud under Windows 95 & Windows 3.x!!!
- Built-in Maildrop Protocol & Nine-Bit Protocol (RS485 appx).
- Compatible with Visual C/C++, Visual Basic, and many more tools.

Features Supported By All Products

- Remap Change baud rates/divisor.
- Product Customization.
- Hardware/Software Flow Control.
- Unlimited number of ports active concurrently.
- 8250/16450/16550 Auto detection/Up to 400 Kbaud.
- 100% Port re-entrant code/Important for multitasking.
- Support all standard dumb multipoint cards.

WCSC Price List

COMM-DRV/Lib (DOS, Win3x, Win95, NT)	\$189.95
COMM-DRV/Lib (Competitive Upgrade)	\$ 99.95
COMM-DRV/VxD	\$189.95
COMM-DRV/Dos	\$ 99.95
4Port Card (16550/8 bits)	\$170.00
4Port Card (16550/16 bits)	\$199.95
8Port Card (16550/16 bits)	\$225.00
8Port Card (16550/16 bits/Ext. Box)	\$299.95
Professional Combo (Lib, VxD)	\$329.95
Software Combo (Lib, VxD, Dos)	\$399.95
Complete Combo (Software 4Port (16550)	\$499.95

WCSC

4425 Kingwood Dr. Suite 21, Kingwood, TX 77339

(800)966-4832 (281)498-4832

(Fax) (281)568-3334 (BBS) (281)568-6401

Internet: sales@wcsnet.com

Supports most intelligent & non-intelligent multipoint cards.

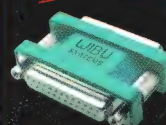
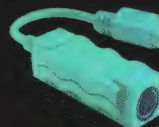


Visa/Mastercard/Discover/AMEX/Checks/Approved P.O. [WWW: http://www.wcsnet.com/home.htm](http://www.wcsnet.com/home.htm)

CIRCLE NO. 600 ON READER SERVICE CARD



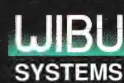
Order your Test-Kit today!
Call now: (800) 986-6578
<http://www.griftech.com>
<http://www.wibu.de>



High Quality Software Protection since 1989

- Secure against systematic "crackers"!
- WIBU®-BOX for LPT, COM and ADB, USB, transparent and cascable, as PCCard and (E)ISA card.
- DOS, Windows™, Windows 95, Windows™, NT, MacOS™, OS/2®, Novell.
- Protection without source code modification
- CD-ROM Protection.
- API independent of programming language and operating system.

NEW:
ISO9001
in Sales,
Production &
Development



WIBU-SYSTEMS AG
Rueppurrer-Strasse 54
D-76137 Karlsruhe, Germany
Tel. +49-721-93172-0 FAX 93172-22
CIS 1001421674 - email: info@wibude



1617 St. Andrews Dr. Lawrence, KS 66047
Tel. (913) 832-2070 FAX (913) 832-8787
CIS 71413624 - email: sales@griftech.com

CIRCLE NO. 602 ON READER SERVICE CARD

Dr. DeeBee® ODBC Driver Kit

The Easiest Way to Create an ODBC Driver

Build an ODBC driver for your database to connect it to
desktop favorites like Access, Visual Basic and Powerbuilder



- Full Source Code Included
- Royalty Free Distribution
- No ODBC Knowledge Required
- 2 to 4 Weeks Development Time



Also available, Dr. DeeBee ODBC Tools

Debug and optimize ODBC applications and drivers fast!

Need More Info? Visit: www.syware.com

Or call: 617 497-1300

—SYWARE— P.O. Box 91 Kendall, Cambridge, MA 02142

CIRCLE NO. 601 ON READER SERVICE CARD

INTRODUCING

Visual Intercept™

"The only project-oriented enterprise bug
tracking system for Microsoft™ Developers"

- Integrates with Microsoft tools & applications
- Enterprise ready & scalable
- 3 tier architecture, ODBC, & 32-bit

1-800-364-5467
Raleigh Group International @ www.ralgi.com



Elsinore
TECHNOLOGIES, INC.™
www.elsitech.com

CIRCLE NO. 603 ON READER SERVICE CARD

Do you need ODBC, OLE-DB, and JDBC access to a proprietary or legacy database?

OpenAccess™
ODBC SDK is the
quickest and
most economical
solution for
your needs!

Your solution will be
ready for distribution
in **weeks**.

ODBC
OLE-DB
JDBC

LAN & Internet

OpenRDA® Server

Proprietary or Legacy
Databases

Ask about
our single
ODBC solution
for heterogeneous
database
access!

Call us! We would be happy to discuss your needs.



Automation Technology, Inc.
2890 Zanker Road, Suite 200
San Jose, CA 95134
Tel 408/473.0200
Fax 408/473.0201

<http://www.atinet.com>

sales@atinet.com

CIRCLE NO. 604 ON READER SERVICE CARD



**Adobe® Developers Association—
The Source for Developer Tools**

Adobe Add powerful support to your application or create
in-demand plug-ins for Adobe Acrobat, Adobe
Illustrator, Adobe PageMaker, Adobe Photoshop,
and Adobe Premiere, with a wide range of available developer toolkits.

To receive a Membership Information Kit, call 1-415-961-4111, or send
a fax to 1-415-967-9231.

For more information, see the Adobe home page on the World Wide
Web at <http://www.adobe.com> and go to the Support & Training section.

Adobe Systems Incorporated, 1585 Charleston Road, Mountain View CA 94039-7900

CIRCLE NO. 605 ON READER SERVICE CARD

CRUSHER!™

**Data
Compression
Toolkits**

The best high performance,
portable **compression
libraries** for DOS, 16-bit
and 32-bit Windows,
OS/2, Unix, Macintosh,
embedded systems, and
practically anything else,
period.

Call 1-800-775-1073

Never shell out to the OS again! Crusher's
robust 45-function API compresses buffers,
files, creates multi-file archives, stores
subdirectories, creates self-extracting
executables, encrypts, and more!



**DC Micro
Development**

P.O. Box 54588
Lexington, Kentucky 40555 USA
Tel (606) 245-4175
Fax (606) 245-9305

info@dcmicro.com <http://www.dcmicro.com> **FREE DEMO**

CIRCLE NO. 606 ON READER SERVICE CARD

"It's
**RedHat
Linux.**"



The most recent, most
stable, Linux products

\$49.95

"Official" Red Hat Linux Boxed Set:
Next generation OS for Intel PCs, Alpha, and SPARC.

\$29.95

Red Hat Linux "Power Tools" & archives:

\$199.00

Applixware Office Suite for Linux:

\$149.00

Red Hat's Motif for Linux:

\$99.95

CDE for Linux Developer's Edition

Client Edition

\$39.95

Books:

Dr. Linux:

\$34.95

Linux Man Pages:

\$34.95

Maximum RPM:

\$34.95

www.redhat.com

tel: 800.454.5502 or 919.572.6500
fax: 919.572.6726

CIRCLE NO. 607 ON READER SERVICE CARD



DriverX™

Hardware Control Library

Easily develop custom hardware control
solutions for NT and Win 95 with no
kernel-mode driver development. C/C++
libs, Delphi unit, and OCX included. PCI
and fast read/write loop support.

DFC

Accelerate your kernel-mode driver development with
the Driver Framework Class library. DFC includes a
driver skeleton generator and C++ classes to help you quickly develop high-
performance VxD, WDM, and NT drivers from common source.

Tetradyne

Software, Inc.

(408)377-6367 FAX: (408)377-6258 sales@tetradyne.com

Download demos from www.tetradyne.com

CIRCLE NO. 608 ON READER SERVICE CARD



Ultimate TCP/IP™

Internet/Intranet Client and Server Development Kit
High Performance C++ Class Libraries

✓ POP3
✓ SMTP
✓ HTTP

✓ FTP
✓ DNS
✓ NNTP

✓ Clients and Servers
✓ Custom Protocols
✓ NT Services

Professional (w / full source code) \$499.00
Includes One year of technical support.

Dundas Software

1 800 463 1492

Sales: 416 239 7472

Fax: 416 239 2183

Email: sales@dundas.com

WWW: www.dundas.com

• 30 day money-back satisfaction guarantee.
• Visa, MasterCard, American Express,
check, money order.
• We will deliver via Internet, Compuserve,
mail or courier.

www.dundas.com

CIRCLE NO. 609 ON READER SERVICE CARD

News Server Technology - FREE Trial

Tired of gaps in news groups because INN can't keep up?

Introducing - DNEWS 4.0

- **Performance** - 300% faster than INN servers. Many more users.
- **Scalability** - Scaling well beyond INN, 0—50 Gig Spool.
- **Easy Installation and Management** - Designed to run unattended, with virtually no administration. Fast to install and setup.
- **Dynamic Feed Option** - Cutting Diskspace and Bandwidth.
- **Advanced Options** - Add usenet or local groups to your web pages with built in News to Web Gateway. **Get the Best!**

Download FREE trial. Visit - <http://netwinsite.com>

CIRCLE NO. 610 ON READER SERVICE CARD

REACH OVER 116,000 PROFESSIONAL SOFTWARE DEVELOPERS FOR JUST PENNIES PER LEAD WITH YOUR AD IN THE PRODUCT DEBUT



Announcing another great game
written with
Fastgraph®
Programmer's Graphics Library
3D Casino Las Vegas
All the excitement of a live
action casino under Windows 95.

More fun than being in Las Vegas!
See the game at <http://www.3d-casino.com>.

Fastgraph programmers write games that ship! DOS, Windows or Windows 95, the games you play were written with Fastgraph.

Fastgraph for DOS: \$249 **Fastgraph for Windows: \$249**
Ted Gruber Software Voice (702) 735-1980
PO Box 13408 FAX (702) 735-4603
Las Vegas, NV 89112
Visit our web page at <http://www.fastgraph.com>

CIRCLE NO. 611 ON READER SERVICE CARD

BAR CODE



Print bar codes within your applications
with our TrueType fonts. Code 39, Code
128, UPC, etc. Free sample code available.

Azalea Software, Inc.
1 800 48-ASOFT www.azalea.com

CIRCLE NO. 500 ON READER SERVICE CARD

C++ LIBRARIES

WMTLIB C++ Multithread Library

Make multithread and multiprocess development simpler with the WMTLIB C++
classes, and gain cross-platform portability at the same time.

WMTLIB includes classes to manage synchronisation and data transfer between
threads, both intraprocess and interprocess, with documented source, royalty
free deployment, and inexpensive licensing per developer.

Supports Win 32 (NT and Win95), OS/2, LinuxThreads, Solaris 2.5 (Solaris
threads and POSIX 1003.1c), UnixWare 2.1, more to follow.

See www.westgold.com for more details.

Westongold Ltd.

email: info@westongold.com or call (+44) 1992 633801

CIRCLE NO. 503 ON READER SERVICE CARD

DEVELOPMENT TOOLS

The McCabe Visual ToolSet™

McCabe & Associates, inventor of the cyclomatic complex-
ity metric, provides visual tools for software testing and
code coverage. Unit and integration test conditions are
generated from C++ code. The tools generate metrics for
polymorphism, encapsulation and inheritance. Overloaded
modules are displayed on the graphical structure charts.

Uncover Your Software  **McCabe Associates®**
For Further Information Call 800-638-6316 or Fax 410-995-1528
<http://www.mccabe.com>

CIRCLE NO. 506 ON READER SERVICE CARD

BOOKS

WROX PRESS

Beginning MFC OLE/Active X Programming from Wrox Press

*Learn how COM works in practice
*Create real world OLE servers & clients
* Step by step projects with full explanations
* Uses MFC & Wizard support for
fast results

For more information about any
Wrox programming titles, please
call 800-USE WROX or 773-
465-3559. When ordering,
Quote code **CL77** for free freight
in USA. Visit our web site for sam-
ple chapters and a full content table.

ISBN: 1-861000-87-7
500 pages \$39.95
Author: Julian Temperman
www.wrox.com

CIRCLE NO. 501 ON READER SERVICE CARD

COMMUNICATION LIBRARIES

16 & 32 bit Serial Comm Libraries

Personal Communications Library Supports 20 ports to
115200 baud, 16 & 32 bit DPMI, multiport dumb cards
(Digiboard, BOCA), any IRQ & UART address, interrupt driven,
HW flow control, 16550 UART & modem AT commands.
Source code included. Specify C/C++ (Win or DOS), Turbo
Pascal (DOS), Visual Basic (Win or DOS), or PowerBASIC
(DOS).

\$75 + s&h includes manuals & one year support. Get **FREE**
shareware version form our BBS or ftp.marshallsoft.com.
Visit our home page at www.marshallsoft.com.

MarshallSoft Computing, Inc.
P.O. Box 4543, Huntsville, AL 35815

email: info@marshallsoft.com
(205) 881-4630 Voice, 880-0925 FAX, 880-9748 BBS

CIRCLE NO. 504 ON READER SERVICE CARD

PARSING JUST GOT EASIER

Parsing input used to mean tricky logic with lots
of places for bugs to hide, slipped schedules,
difficult maintenance. Nobody's idea of fun.

Now, more powerful, more convenient,
AnaGram V1.5 changes all that. You can take
charge and concentrate on the real problems.
Expressions? Database queries? Scripts?
Command languages? No problem. *Call for free
trial copy.*

Anagram™ by Parsifal Software
P.O. Box 219, Wayland, MA 01778

CIS: 72603,1763 info@parsifalsoft.com
(800) 879-2577 Voice/Fax (508) 358-2564

CIRCLE NO. 507 ON READER SERVICE CARD

C++ COMPILERS

CAD-UL
Professional
C++
Embedded
Compilers
&
Dev. Tools
<http://www.cadul.com>

CIRCLE NO. 502 ON READER SERVICE CARD

CONSULTING

**LET US LEAD YOU THROUGH THE
DATA MINE**
AHC AMERICAN
HEURISTICS
CORPORATION
Consulting support and training
in business intelligence
www.heuristics.com • info@heuristics.com
304-547-4201

CIRCLE NO. 505 ON READER SERVICE CARD

Btrieve Database Access Fast & Easy!

Smithware
Get blinding performance and native access to Btrieve
databases without writing code! Our advanced data
binding technology outperforms the VB data control,
ODBC, Jet and other access methods. Smithware's
ActiveX Controls give you total control over your data
with seamless integration of Btrieve's client/server
extended operations, extended joined record sets,
bound controls and more. Use our controls in your favorite development includ-
ing VB, Delphi, VC++, Internet Explorer or in any environment that supports
ActiveX. You can even enable your application over the web. Check out our
benchmarks and Internet demo at www.smithware.com/activex or call us
today and see for yourself how you can get fast easy access to your Btrieve Data!

Smithware, Inc.
2416 Hillsboro Road, Suite 201, Nashville, TN 37212
Phone: 615-386-3100 or 800-837-6316
Email: info@smithware.com
<http://www.smithware.com>

CIRCLE NO. 508 ON READER SERVICE CARD

DEVELOPMENT TOOLS

Boost your Basic!

Develop your app faster. Get TeraTech tools! 36 VB, QB, PB, C & Xbase libraries available. Call us and we'll mail you a free demo & booklet ASAP. Call now!

60 Day MBG!

- * Dazzle/VB - images
- * VLibe - print/comm
- * ProMath/VB - numerics
- * VLib/VB - financial
- * VoxLibPro - telephony
- * SpellCheck/VB - spelling
- * ProBas - & other 15 DOS libs
- * VB training - 2 day intro class
- * Custom - C, ASM, VB programming

800-447-9120 ext. 1280

Dept 1280, 100 Park Avenue, Suite 360, Rockville MD 20850
Int'l: 1-301-424-3903 Fax: 301-762-8185 BBS: 301-762-8184
info@teratech.com http://www.teratech.com/teratech/

CIRCLE NO. 509 ON READER SERVICE CARD

Programming Tools For Programmers

VB • DELPHI • CLIPPER
INTERNET • VdBASE • VO

- Product Info
- Demos
- Online Ordering
- Online Delivery

Get Connected!
www.zaccatalog.com

CIRCLE NO. 510 ON READER SERVICE CARD

Zip and Unzip from your apps with the XCEED ZIP COMPRESSION LIBRARY

- Easy to use VBX, OCX and VCLs
- Fast, compact and reliable
- 100% PKZip 2.04g compatible
- All 16 & 32-bit controls for \$199.95
- Over 40 Zip and Unzip functions!

Get the fully functional free trial version!
www.xceedsoft.com/dobbs

Xceed Software 1-800-865-2626 1-514-442-2626

CIRCLE NO. 511 ON READER SERVICE CARD

C++ to HTML

- Generates useful docs for C/C++ code
- Uses existing comments with no changes
- Metrics
- HTML and RTF
- Windows 95 and Windows NT
- **FREE** evaluation version at **www.bbeesoft.com**

OBJECT OUTLINE® 2

1-800-214-4746, *297
info@bbeesoft.com, P. O. Box 541, Hudson, MA 01749

CIRCLE NO. 512 ON READER SERVICE CARD

ADAPTIVE OBJECT STORAGE™

Need a more flexible storage model than simple persistent C++ classes provide? Adaptive Object Storage provides a superior solution to managing data structures as they change over the life of an application.

Features:

- compressed streamable databases
- access to most data before database is restored
- load-on-demand for large values
- no I/O coding required
- Java support

Great for bandwidth-efficient and responsive client-server Web applications. For C++ applications on Win32 platforms.

Galapagos
Software, Inc.

Get a **FREE EVALUATION** at:
http://www.galapagos1.com
For more info, call 800-562-2895

CIRCLE NO. 513 ON READER SERVICE CARD

REACH OVER 116,000

professional software
developers for just pennies
per lead with your ad in the
Marketplace.

EDUCATION

ADDRESS FOR SUCCESS
<http://www.aics.edu>

- Earn B.S. and M.S. in Computer Science
- DISTANCE EDUCATION
- Object oriented B.S. program
- New courses in Java, Networking, HTML, MIS
- Approved by more than 275 companies
- Follows ACM/IEEE guidelines
- Thousands of students throughout U.S.

Free catalogue 1-800-767-AICS
or <http://www.aics.edu>

ENCRYPTION



When you need to provide industrial
strength encryption within your software
projects, simply wire in carrick.

1 800 ENCRYPT

www.encryption.com

CIRCLE NO. 514 ON READER SERVICE CARD

FORTRAN

The SPINDRIFT Library Version 3.0

The Complete FORTRAN Programmer's Toolkit.

Create sophisticated state-of-the-art user interfaces for your FORTRAN programs. Spindrift's callable subroutines and functions let you

- Have data entry screens, dialog boxes, scrolling list boxes, pull-down menus, push buttons, help panels, etc. All with full mouse support.
- Much more! 281 subroutines and functions for keyboard, screen, mouse, DOS control, and hardware status.

Comprehensive demonstration program shows what you can do and how to do it.

Now shipping Version 3.0 with all new User Guide.

PRICE: 16 Bit Compilers - \$149 32 Bit Compilers - \$289

Spindrift Laboratories, Ltd.

8 Tanglewood Path • Galena • IL 61036 • USA
Phone: (815) 777-8240 • FAX: (815) 777-8241
E-mail: dgable@galenalink.com

CIRCLE NO. 515 ON READER SERVICE CARD

GRAPHIC LIBRARIES



Image Processing Library

Fast BMP, TIFF, PCX, GIF, TGA, PNG, JPEG. Adjust brightness, contrast, sharpen, create filters, resize, rotate, + more of single image, multiple images, or any image area; color reduction to optimum, specific, or std. palette; print; scan; crop; combine; compare; blend images.

DOS \$199, 16-bit DLL \$299, 32-bit DLL \$499

Catenary Systems
314-962-7833/fax: 314-962-8037
www.catenary.com/victor

ask for free demo src avail visa/mc/c.o.d.

IMAGING LIBRARIES

CORPORATE SUITE
ImageLib VCL
Winner of 3 Awards!

"The Most Robust Document
Imaging Tool In Today's Market."

- Easy to Use
- Royalty-Free
- Ideal for Document Imaging
- Ultimate Scanner Interface
- 25 Formats
- Image Manipulation Package

free trial versions
imagelib.com/corp.html
(818) 766-3900
SkylineTools

New C++ Builder
Delphi / C / VB

CIRCLE NO. 516 ON READER SERVICE CARD

INTERNET

WEB COMMUNICATIONS MADE EASY!

App-Link Internet Messaging System

Vis C++ VB MS IE Excel FoxPro The WEB Delphi Optima Access

The new App-Link Internet Messaging System (IMS) makes it EASY to communicate between applications locally, or over any TCP/IP network. With IMS, all you have to do is say "here's my data, here's where I want it to go", and send a receive event fires on the other end when the message is delivered, passing the data and its origin as parameters. Communicate between any containers that support a VBX or OCX. All versions of Windows are supported, and other features such as queuing, message packing and formatting, auto program load, etc are also included. Forget TCP/IP protocols, DCOM, etc. App-Link IMS is fast, reliable and above all else, extremely easy to use. App-Link IMS handles the dirty networking tasks so you can relax and focus on your application. Please visit our web site for details.

http://www.synergysw.com

Synergy Software Technologies Inc. sales: 800-294-8514, tel: 802-878-8514
159 Pearl Street fax: 802-878-4055
Essex Junction, Vermont 05452 **sst@synergysw.com**

CIRCLE NO. 517 ON READER SERVICE CARD

LIBRARIES

Add ZIP (and UNZIP too!) to your Windows applications!

DynaZIP™ 3.0 Data Compression Toolkits

The new ROYALTY-FREE DynaZIP family of developer's tools let you add ZIP and UNZIP capabilities to your Windows applications. No more "shelling" to DOS, no more fussing with proprietary compression formats. DLLs, VBXs, OCXs and a new database interface provide full access from many languages. Fast, reliable, and easy to use! 16 and 32 bit versions, supports long filenames.

Fully supported, 30-day no-risk guarantee!
Call today, toll free: (800) 962-2949

Inner Media, Inc., Hills, NH USA (603) 465-3216, fax (603) 465-2195
Visit our web site at <http://www.innermedia.com>

CIRCLE NO. 518 ON READER SERVICE CARD

Complete C++ Math Library

- Complete multidimensional array language
- Complete package of mathematic
- Full error handling package

Advanced Support Modules

Signal Processing (NEW) Finite Element
Financial Utility (NEW) Least Squares
Call for complete list

FREE demo program on website

http://www.wolfenet.com/~dyad dyad@wolfenet.com
Dyad Software Corporation
6947 Coal Ck Pkwy SE #361 Newcastle WA 98059
(206) 637-9426 (800) 366-1563

CIRCLE NO. 519 ON READER SERVICE CARD

PRODUCTS WANTED

WANTED

Major publisher with national
distribution seeks SOHO, Personal
productivity titles written in C+ or
C++ for Win 3.1, NT, Workgroups,
Win 95 for immediate license
or affiliate relationship.

Fax Ms. Zack 1-954-698-0057

CIRCLE NO. 520 ON READER SERVICE CARD

Add Intelligent Components

Diagnose, advise, configure, evaluate, and plan with the Amzi[®] LogicServer[™] tools and libraries (DLLs) for Win NT 95 3.x, Linux & Solaris. For C/C++, Java, VB, Delphi, Web servers and more. Use ODBC data. **FREEWARE**, demos and articles on our Web Site.

Amzi[®] inc. Email info@amzi.com
40 Samuel Prescott
Stow, MA 01775 USA Call +1-508-897-7332
Fax +1-508-897-2784

CrypKey Software Licensing System

"Software protection with NO hardware lock and NO disk key"

CrypKey is software copy protection that is:
• completely secure from any disk copy program
• perfect for CD-ROM or INTERNET distribution!
• cost effective, user friendly, and 100% guaranteed to satisfy!

CrypKey can increase your software sales:
• upsell options and levels of your software
• lease or demo your software by runs or time
• enable or upgrade your customers instantly by phone, fax or email

New! unique Ready-To-Try features upon install allows 1 trial period only per customer.
New! unique Add-On feature-add more options, levels, runs or time to existing licenses.
New! CrypKey Instant-protects in just 5 minutes with no source code changes.

CrypKey is completely compatible with MS-DOS, MS-Windows 3.x, Win32s, Win95, Win NT, and manages network licenses on all Novell and Microsoft operating system based networks. CrypKey is produced by Kenonic Controls Ltd.—engineering and software since 1972.

Kenonic Controls Limited
7175 - 12th Street South East
Calgary, Alberta, Canada T2H 2S6
(403) 258-6200 Fax: (403) 258-6201
INTERNET: crypkey@kenonic.com
WEB: <http://www.kenonic.com/crypkey.html>

CIRCLE NO. 523 ON READER SERVICE CARD

Lock Out Pirates! Lock In Profits!

The Marx CRYPTO-BOX is the result of over 10 years experience in effective software protection.

- microprocessor controls ID codes, memory, dynamic algorithm and high speed data encryption
- remote access to passwords and counters
- license metering in networks: single key per LAN

MARX
INTERNATIONAL, INC.
404-321-3020 FAX: 404-321-0760
1-800-MARX INT
20 Executive Park West, Suite 2027, Atlanta, GA 30329
Visit our Home Page: <http://www.marx.com>

CIRCLE NO. 526 ON READER SERVICE CARD

SW DEVELOPMENT KITS



DLL - OCX - VBX

- Barcode recognition from bitmap
- Barcode + text write into bitmap
- Skew correction
- Noise + Crop
- Annotate image
- Line removal
- TIFF r/w - de/compress - fast scale

Axtel, Inc. Tel: (714) 964-6666
Fax: (714) 964-6766

Free demo on <http://www.axtel.com>

CIRCLE NO. 527 ON READER SERVICE CARD

TOOLS

C and C++ DOCUMENTATION TOOLS (v. 7.0)

- **C-CALL (\$69)** Graphic-tree of caller/called function hierarchy, cross-reference, file/function index.
- **C-CMT (\$69)** Creates/inserts/updates comment-blocks (functions/identifiers used) for each function.
- **C-METRIC (\$59)** Calculates path complexity, counts lines with comments, code, 'C' statements.
- **C-LIST (\$69)** Lists and action-diagrams, or reformats source into user-selected standard formats.
- **C-REF (\$69)** Creates cross-reference of local/global/define/parameter identifiers.
- **C-DOC (\$199) PACKAGE** All 5 programs integrated as DOS program. <10,000 lines. C-BROWSE Windows graphic-tree viewer.
- **C-DOC Professional (\$299)** DOS, Windows, OS/2, 1,000,000+ lines.
- **NEW VER 7.0! WEB HTML REPORTS!**

SOFTWARE BLACKSMITHS INC. email @ swbs.com
6064 St Ives Way, Mississauga Voice/Fax (905) 858-4466
ONT Canada L5N-4M1 <http://www.swbs.com>

CIRCLE NO. 528 ON READER SERVICE CARD

Az-Tech Software

Internet:
www.az-tech.com
E-Mail:
sales@az-tech.com
Voice:
(816) 776-2700
Fax:
(816) 776-8398

AZ Leaders in
Software
Security

STOP PIRACY NOW



**Copy Protection
License Control
Internet Distribution**

CIRCLE NO. 521 ON READER SERVICE CARD

HASP. Protect Your Software. Increase Your Revenues.

This year software piracy will cost software developers billions of dollars. Why be a victim when HASP – the industry's most effective protection solution – is there to protect your software! Rated #1 by the NSTL, HASP will increase your revenues, monitor network licensing and much more while keeping the pirates away! **Call now to order your low-cost HASP Developer's Kit!**

North America Tel: 800-223-1277, 212-564-5678,
Fax: 212-564-5377, Email: hasp.sales@us.aks.com
Int'l Office Tel: +972 3 636-2222,
Fax: +972 3 537-5796, Email: hasp.sales@aks.com

ALADDIN
The Professional's Choice

CIRCLE NO. 524 ON READER SERVICE CARD

LockSmith

Disk, CD or Internet installed software

- ✓ Date & Time Locks
- ✓ Feature Switching
- ✓ Network License
- ✓ Data Encryption
- ✓ Remote Update
- ✓ Serialization

Secure your software disks with:
No royalties or limit on products

1-800-688-8116

www.LockSmithShop.com
Sales@LockSmithShop.com

CIRCLE NO. 522 ON READER SERVICE CARD

The New Level in Security

- SONY Fingerprint Identification Units -

- Completely self-contained fingerprint scanner with internal CPU + RS-232 interface
- Enables user identification and authentication via fingerprints
- Can store 1000 templates
- Includes software developer's kit for Windows and DOS



Call or write for FREE brochure
See our web site for detail

I/O Software, Inc. 1533 Spruce St. Riverside, CA 92507
(909) 222-7600 www.iosoftware.com

CIRCLE NO. 525 ON READER SERVICE CARD

DR. DOBB'S WEB DIRECTORY

CASE/TESTING & DESIGN

- **PURE ATRIA** <http://www.pureatria.com>

CD ROM

- **INFOMAGIC** <http://www.infomagic.com>

CROSS PLATFORM/GUI DEV. TOOLS

- **SOFTTEL VDM** <http://www.softtelvdm.com>

- **GAMELON by MENAI** <http://www.menai.com>

- **BRISTOL TECHNOLOGY** <http://www.bristol.com>

"With Wind/U, you can use the Microsoft Win32 API and MFC to develop and maintain a single source code for Windows, UNIX, OPEN VMS, and OS/390 versions of your applications."

DOCUMENTATION TOOLS

- **Willows Software** <http://www.willows.com>
Use The Willows Toolkit for porting 16- and 32-bit Windows[™] applications to UNIX, Macintosh, real-time, network and NC operating systems, JAVA, and other platforms.

EMBEDDED SYSTEMS

- **WIND RIVER SYSTEMS** <http://www.wrs.com>

- **Ristanovic Case**
URL: "<http://www.restancase.ch/da>"
Strong WEB Site for C or PL/M Developers

LANGUAGES/DEVELOPMENT TOOLS

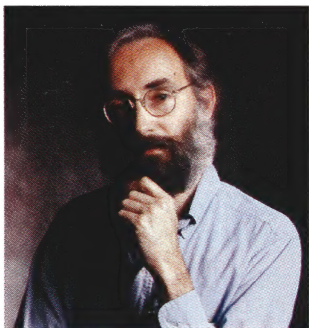
- **INTEL** <http://www.intel.com/ial/vtune/index/html>
- **INTERACTIVE SOFTWARE ENGINEERING** <http://www.eiffel.com>
- **QUINTESSOFT ENGINEERING, INC.** <http://www.quintessoft.com>
Code navigator for C++

MISCELLANEOUS

- **SYMANTEC** <http://www.careermosaic.com/cm/symantec/home.html>

UTILITIES

- **NuMEGA** <http://www.numega.com/>



Perplexed

My world consists of those items that I've filed away in their proper places, and all that other stuff. It has taken me altogether too long to figure out that nearly all the interesting stuff is in the second category, the unfiled mess. Properly filed things *become* less interesting through the very act of filing them. And the *most* interesting things are the things that I couldn't file if I tried.

I refer to the nonplace where I file these unfiled items as my futility closet, because of the futility of trying to find anything there. All my best search strategies are directed at finding filed things; I have almost no algorithms for sifting through the muddle in my futility closet. But there should be some algorithm I could use, other than blind stumbling search. There's really a lot that I know about this uncategorized category.

For one thing, everything there is there because I once thought it was interesting or useful. That narrows it a lot. And these things have a lot of connections to one another and even to items that are properly filed. They just don't fit well into any of my tidy hierarchies. I think the answer has something to do with being precise about uncertainty, but I don't know precisely what. I'm perplexed.

That could be a clue right there. Ted Nelson once defined the word "perplex" (apparently coined by Frederick C. Crews in *The Poo Perplex*) as "a precisely delineated muddle." Precision about uncertainty. Ted thinks that one of the many virtues of hypertext is that it allows "the true uncertainties of thought to be accurately recorded."

Actually, I thought that was the definition of this column.

New subject. I mentioned here recently that I was setting up a Hall of Flame web site to which anyone can submit nominations for particularly dumb things that deserve recognition.

My inspiration was an old magazine by Roger Price called *Grump*, with the subtitle "For people who are Fed Up with all the Dumb Things that are going on." (I'm doing the capitalization from memory.) The site is up at <http://www.cruzio.com/~mswaine/>. It's getting some good dumb-thing contributions, and I'll share some of the dumbest here from time to time.

I hesitate to suggest this, but since I already did (see my "Programming Paradigms" column elsewhere in this issue), I guess it's too late for reconsideration. If you have written or come across any song parodies that deal with programming themes, send them my way. I haven't created a web page for them yet, and I don't know for sure that I will, but that's the idea. My hesitation comes from my poetic sensitivity. I can't write poetry, can hardly carry a tune, but I was cursed with an ear for meter. Poetry that tries to follow some meter and fails hurts my ears. I suspect I'm inviting some pain. We'll see.

By the way, Reid Byers offers a more high-tone theme for my Hall of Flame web site:

I think what Roger Price was talking about (and I remember *Grump* fondly) is the same thing Barbara Tuchmann wrote of in her wonderful book, *The March of Folly*. Tuchmann defines folly as people acting against their own self-interest. Her historic examples include The Trojan Horse, England during the American Revolution, the U.S. in Vietnam. Thought this might make a good working definition for your project.

The March of Folly. Cool. Thanks, Reid.

Michael Swaine

Michael Swaine
editor-at-large
mswaine@cruzio.com

— NEW —
ActiveX
Support

Fast Database Engine

Now discover CodeBase 6—The fastest and smallest database engine available with xBASE file compatibility!

Query a million records in just 0.97 of a second,
or add ten thousand records in just 2.17 seconds.

Discover these valuable benefits of CodeBase:

- C, C++, Visual Basic, Delphi and Java are all supported
- FoxPro, Clipper and dBASE file support, including full multi-user compatibility
- Runs under Windows 95, NT, 3.1, DOS, Mac, OS/2, Solaris, SunOS, AIX, SCO, Linux, UnixWare...
- Tiny libraries help apps load quickly and use less memory
- 100% royalty-free client/server configuration included
- Transaction processing and logging
- Crash recovery is built-in
- Full featured report writer included
- Source code included
- Royalty free distribution

New ActiveX

Seamlessly integrate our fast database engine with your favorite compiler using our new ActiveX database support.



Award-Winning performance 5 years in a row!

“CodeBase gives ACT! the fast database access contact management users need”

- Michael Plasterer, Director of Development,
Symantec



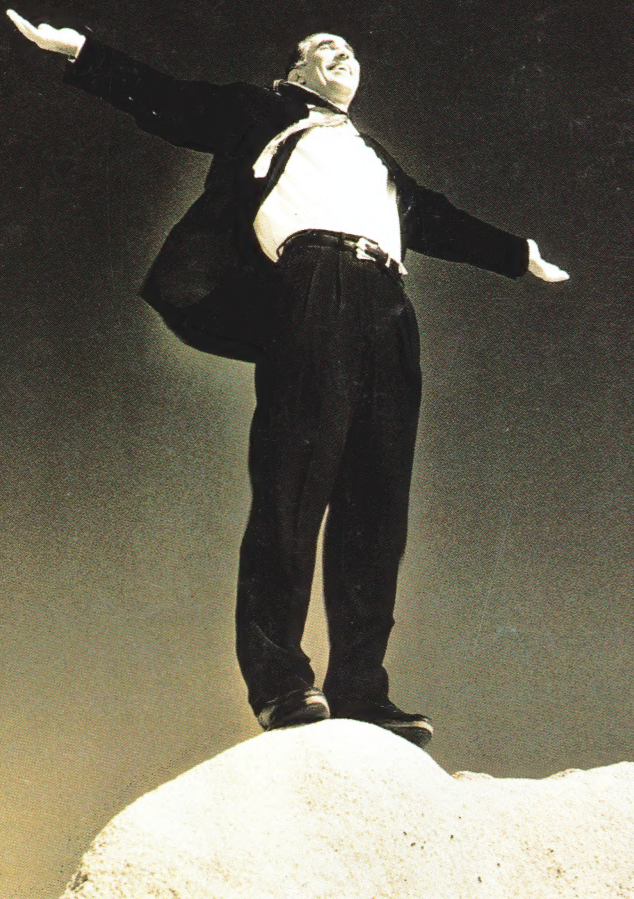
The Database Engine for Programmers

FREE 30 Day Test Drive

Test drive the new CodeBase 6 for 30 days with your own code. No risk. No obligation. Order yours today.

Call: 403-437-2410

SEQUIER SOFTWARE INC.
Fax: 403-436-2999
Email: info@sequier.com
P.O. Box 783 Greenland NH 03840 Web site: www.sequier.com



ON TIME. ON BUDGET. ON DELPHI.

DELPHI 3. THIS CHANGES EVERYTHING.

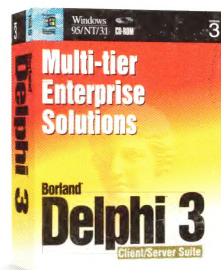
Is your development too complex? Are your budgets tight? Schedules impossible? People overworked? Do you depend on an increasing number of applications to run your business, talk to your customers, link in suppliers? Now you can turn all your problems into opportunities—with Delphi™ 3.

Delphi 3 adds more than 50 significant technologies to help you make effective business decisions, scale with growth, get products to market quickly, and increase overall productivity. Delphi 3 innovations include:

- One-step ActiveX creation for enterprise-wide reusable components
- BusinessInsight for integrated Decision Support development
- CodeInsight Wizards for ease-of-use and increased productivity
- MIDAS™ Broker technologies for 24x7 business-critical applications
- WebBroker for HTML delivery of database information to any browser on any client
- High-performance native database drivers for Oracle, Sybase, Informix, DB2, AS/400, InterBase®, MS SQL Server, and MS Access.

Delphi continues to set the standard for high productivity, rapid development of high-performance client, server, and multi-tier applications for the enterprise, individual, and Internet.

Get *today* what other tools are promising tomorrow. Call Borland at 1-800-336-6464, ext. 52019, or visit us at www.borland.com/delphi/



Borland
www.borland.com